یونٹ 7

کمپیوٹیشنل تھنکنگ

اس باب کا مطالعہ کرنے کے بعد طلبہ اس قابل ہو جائیں گے وہ:

- کمپیوٹیشنل تھنکنگ اور اس کے کلیدی اجزا کی وضاحت کر سکیں: پرابلم بریک ڈاؤن، پیٹرن کی شناخت، ایبسٹرکشن اور الگورتھم۔
- کمپیوٹیشنل تھنکنگ کے اصولوں کی وضاحت کر سکیں، بشمول مسئلہ کی تفہیم، مسئلہ کو آسان بنانا، اور حل کا انتخاب اور ڈیزائن۔
- الگورتھم ڈیزائن کے طریقوں، خاص طور پر فلو چارٹ اور سوڈوکوڈ کی وضاحت کریں، اور ان کے مابین اختلافات کو سمجھ سکیں۔
- الگورتھم کو بصری طور پر پیش کرنے کے لیے فلو چارٹ بنائیں اور ان کی تشریح کر سکیں۔
- ایک منظم، انسانی پڑھنے کے قابل فارمیٹ میں الگورتھم کا خاکہ پیش کرنے کے لیے ''سوڈوکوڈ'' لکھ سکیں۔
- الگورتھم سرگرمیوں میں مشغول رہیں جیسے ڈیزائن اور تشخیص کی تکنیک۔
- ان کی درستی کی دستی طور پر تصدیق کرنے کے لیے فلو چارٹ اور سوڈوکوڈ کے ڈرائی رن کریں۔
- LARP کے تصور اور اہمیت کو سمجھ سکیں (منطق کی منطق) مسائل کے حل کے لیے الگورتھم۔
- لکھنے کے الگورتھم اور ڈرائنگ فلو چارٹ کی مشق کرنے کے لیے LARP سرگرمیوں کو نافذ کر سکیں۔
- الگورتھم میں مختلف قسم کی غلطیوں کی نشاندہی کر سکیں، بشمول سنٹیکس کی غلطیاں، منطقی غلطیاں، اور رن ٹائم غلطیاں۔
- الگورتھم میں غلطیوں کو تلاش کرنے اور ٹھیک کرنے کے لیے ڈیبگنگ تکنیک کا اطلاق کر سکیں۔
- LARP کے دوران پیش آنے والے عام غلطی کے پیغامات کو پہچان کر ان کو حل کرنے کا طریقہ سیکھ سکیں۔
- حقیقی دنیا کے منظرنامے پر کمپیوٹیشنل تھنکنگ کے اصولوں کو لاگو کر کے مسئلہ حل کرنے کی مہارت کا مظاہرہ کر سکیں۔
- مختلف الگورتھم کی کارکردگی کا اندازہ کر سکیں اور کارکردگی کے تجزیہ کی بنیاد پر انھیں بہتر بنا سکیں۔

انشائی طرز سوالات

محکمہ تعلیم کی نئی امتحانی تکنیکس (Knowledge, Understanding, Application, Analytical & Conceptual) کی روشنی میں مرتب کیے گئے انشائی طرز سوالات

7.1 کمپیوٹیشنل تھنکنگ کی تعریف

سوال 1: کمپیوٹیشنل تھنکنگ کی تعریف کریں نیز ڈی کمپوزیشن کے عمل کی ایک مثال کے ساتھ وضاحت کریں اور اس کی اہمیت بیان کریں۔

جواب: کمپیوٹیشنل تھنکنگ (CT) کسی مسئلہ کو حل کرنے کا ایک عمل ہے جس میں پیچیدہ مسائل کو اس طرح حل کرنے کے لیے مہارتوں اور تکنیکوں کا ایک مجموعہ شامل ہے جس پر کمپیوٹر بھی عمل درآمد کر سکتا ہے۔ اس طریقہ کار کو کمپیوٹر سائنس سے باہر بھی مختلف شعبوں میں استعمال کیا جا سکتا ہے، جیسے بیالوجی، ریاضی اور یہاں تک کہ روزمرہ کی زندگی۔

ڈی کمپوزیشن ایک طریقہ ہے جس میں کسی پیچیدہ مسئلے کو چھوٹے، آسان حصوں میں تقسیم کیا جاتا ہے۔ یہ کمپیوٹیشنل تھنکنگ کا ایک اہم حصہ ہے کیونکہ اس سے مشکل مسائل کو چھوٹے حصوں میں بانٹ کر زیادہ مؤثر طریقے سے حل کیا جا سکتا ہے۔

مثال: اگر ہم ایک پرندوں کے لیے گھونسلہ بنانے کا کام کریں تو یہ مشکل لگ سکتا ہے، لیکن اگر ہم اسے چھوٹے حصوں میں تقسیم کریں تو یہ آسان ہو جائے گا۔

https://stepacademyofficial.com/

گھونسلہ بنانے کے مراحل درج ذیل ہو سکتے ہیں:

1- ڈیزائن بنانا: گھونسلے کی شکل، سائز اور ڈیزائن طے کریں۔

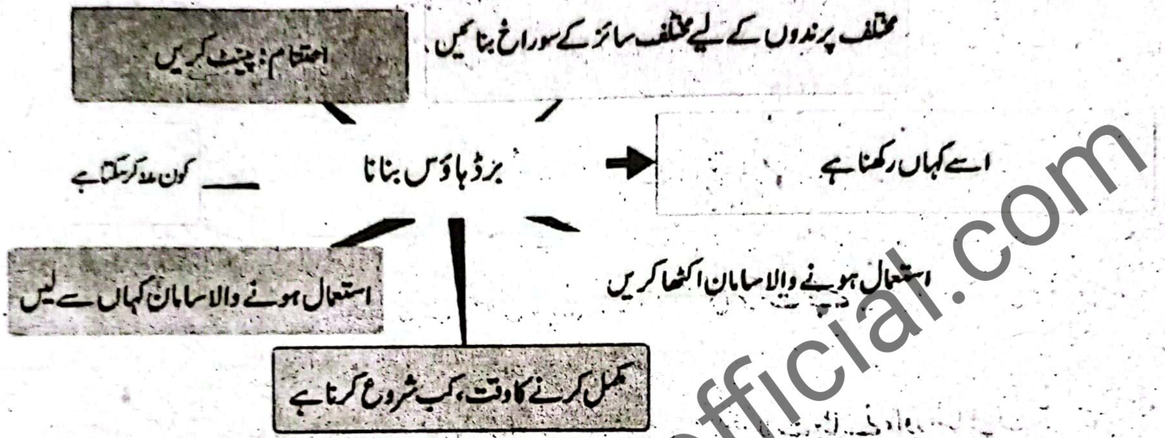
2- مواد اکٹھا کرنا: لکڑی، کیل، رنگ، اور اوزار جیسے ہتھوڑی اور آری جمع کریں۔

3- لکڑی کاٹنا: لکڑی کو مطلوبہ سائز میں کاٹیں۔

4- جوڑنا: تمام حصوں کو ترتیب سے جوڑ کر گھونسلہ بنائیں۔

5- رنگ کرنا: گھونسلے کو رنگ کریں اور خوبصورت بنائیں۔

6- لگانا: مناسب جگہ پر گھونسلہ نصب کریں تاکہ پرندے اسے استعمال کر سکیں۔



پرندوں کے گھر کی تعمیر

اہمیت: ڈی کمپوزیشن خاص طور پر سافٹ ویئر ڈویلپمنٹ میں بہت اہم ہے، جہاں بڑے پروجیکٹس کو مختلف حصوں میں تقسیم کیا جاتا ہے، جیسے یوزر انٹرفیس، بیک اینڈ ڈویلپمنٹ، اور ڈیٹا بیس مینجمنٹ۔ اس طریقے سے مختلف افراد یا ٹیمیں الگ الگ کام کر سکتی ہیں، جس سے کام کی رفتار اور کارکردگی بہتر ہوتی ہے۔ یہ طریقہ مسائل کو حل کرنے کے لیے ایک مؤثر حکمت عملی فراہم کرتا ہے اور پیچیدہ مسائل کو آسان بناتا ہے۔

سوال 2: پیٹرن کی شناخت کیا ہے؟ اور یہ مربعوں کے رقبے معلوم کرنے میں کیسے مدد دیتا ہے؟ ایک مثال دیں۔

جواب: پیٹرن کی شناخت ایک ایسا طریقہ ہے جس میں ہم مختلف مسائل یا ڈیٹا میں مماثلتیں اور ترتیب کو پہچانتے ہیں۔ یہ کمپیوٹیشنل تھنکنگ کا ایک اہم حصہ ہے جو مسائل کو بہتر طریقے سے سمجھنے میں مدد دیتا ہے۔

مثال: فرض کریں ایک طالب علم اکثر پیر کے دن ہوم ورک کرنا بھول جاتا ہے۔ اگر وہ اس بات کو نوٹ کرے اور اتوار کو ایک یاد دہانی لگائے تو وہ اپنی غلطی کو درست کر سکتا ہے۔

اسی طرح، پیٹرن کی شناخت ریاضی میں بھی مددگار ثابت ہوتا ہے، جیسے مربعوں کے رقبے نکالنے میں۔

مربع کے رقبے نکالنے کے لیے ہم یہ پیٹرن دیکھ سکتے ہیں:

- سائیڈ 1: رقبہ=1
- سائیڈ 2: رقبہ=1+3=4
- سائیڈ 3: رقبہ=1+3+5=9
- سائیڈ 4: رقبہ=1+3+5+7=16
- سائیڈ 5: رقبہ=1+3+5+7+9=25

یہاں ہم دیکھ سکتے ہیں کہ مربع کے رقبے میں اضافہ مسلسل طاق نمبروں کو جمع کرنے سے ہوتا ہے۔ اس پیٹرن کو سمجھنے سے ہم بغیر ضرب لگائے بھی مربعوں کے رقبے نکال سکتے ہیں، جو مسئلے کو جلدی اور آسانی سے حل کرنے میں مدد دیتا ہے۔

	+1	+1	+1	+1	+1	+1	
Side	1	2	3	4	5	6	7
Area	1	4	9	16	25	36	47
	+3	+5	+7	+9	+11	+13	

stepacademyofficial.com

سوال 3: تجدید کیا ہے؟ اور یہ مسائل کے حل میں کیسے مددگار ثابت ہوتا ہے؟ ایک مثال دیں۔

جواب: تجدید کی تعریف: ابسٹرکشن ایک ایسا طریقہ ہے جس میں غیر ضروری تفصیلات کو ہٹا کر صرف ضروری معلومات پر توجہ دی جاتی ہے۔ اس سے مسائل کو بہتر طریقے سے سمجھنا اور حل کرنا آسان ہو جاتا ہے۔

تجدید مسائل کے حل میں کیسے مدد کرتا ہے؟

-1 مشکل کو آسان بناتا ہے: غیر ضروری تفصیلات نکال کر مسئلہ کو سمجھنے میں آسانی ہوتی ہے۔

-2 فوکس کو بہتر بناتا ہے: ہم صرف اہم چیزوں پر دھیان دے سکتے ہیں اور غیر ضروری باتوں سے بچ سکتے ہیں۔

-3 مسئلے کو چھوٹے حصوں میں تقسیم کرتا ہے: پیچیدہ مسائل کو آسانی سے حل کرنے میں مدد ملتی ہے۔

-4 کام کی رفتار بڑھاتا ہے: جب صرف ضروری باتوں پر توجہ دی جائے تو حل جلدی اور مؤثر طریقے سے نکل سکتا ہے۔

مثال: چائے بنانے کا عمل ایک عام مثال ہے۔ اگر ہم اس عمل کو تفصیل سے دیکھیں تو بہت سی غیر ضروری باتیں بھی شامل ہو سکتی ہیں، لیکن تجرید کے ذریعے ہم اسے آسان مراحل میں تقسیم کر سکتے ہیں:

-1 پانی ابالیں۔ -2 چائے کی پتی یا ٹی بیگ ڈالیں۔

-3 چند منٹ انتظار کریں۔ -4 کپ میں ڈال کر دودھ اور چینی شامل کریں۔

اس عمل میں ہم نے چولہے کے کام کرنے کے اصول یا پانی کے مالیکیولز کی حرکت جیسے غیر ضروری عوامل کو شامل نہیں کیا۔ صرف ضروری معلومات پر توجہ دی گئی، جس سے مسئلہ سمجھنا اور حل کرنا آسان ہو گیا۔

نتیجہ: ابسٹرکشن ایک مؤثر طریقہ ہے جو پیچیدہ مسائل کو آسان بنانے، غیر ضروری تفصیلات ہٹانے، اور مسائل کے حل میں مدد فراہم کرتا ہے۔ اس کی مدد سے ہم بہتر اور زیادہ مؤثر طریقے سے کام کر سکتے ہیں۔

سوال 4: الگورتھم کی وضاحت کریں اور حقیقی زندگی کی مثالوں سے اس کی اہمیت کو بیان کریں۔ ساتھ ہی یہ بھی بتائیں کہ الگورتھم مسائل کو مرحلہ وار کیسے حل کرتا ہے؟

جواب: الگورتھم ہدایات کا ایک ترتیب وار مجموعہ ہوتا ہے جس کا استعمال کسی مسئلے کو حل کرنے یا کوئی کام مکمل کرنے کے لیے کیا جاتا ہے۔ یہ بالکل اسی طرح ہے جیسے کسی کیک بنانے کی ترکیب کو مرحلہ وار فالو کیا جاتا ہے تاکہ صحیح نتیجہ حاصل ہو۔ الگورتھم وہ درست ترتیب فراہم کرتا ہے جس کے ذریعے کسی کام کو آسانی سے سمجھا اور مکمل کیا جا سکتا ہے۔

حقیقی زندگی میں الگورتھم کی مثالیں

-1 کیک بنانا

کیک بنانے کی ترکیب ایک الگورتھم کی مثال ہے کیونکہ اس میں مخصوص ہدایات دی جاتی ہیں جن پر عمل کرنا ضروری ہوتا ہے، جیسے:

(i) اوون کو پہلے سے گرم کرنا (ii) اجزاء کو ناپنا اور جمع کرنا

(iii) تمام اجزاء کو مکس کرنا (iv) بیٹر کو گریس کیے ہوئے برتن میں ڈالنا

(v) مخصوص وقت تک بیک کرنا (vi) مکمل ہونے پر اوون سے نکالنا

اگر ان مراحل کو درست ترتیب میں نہ اپنایا جائے (مثلاً اوون کو پہلے سے گرم نہ کیا جائے یا اجزاء کو صحیح طریقے سے مکس نہ کیا جائے)، تو کیک ٹھیک سے نہیں بنے گا۔

-2 پودا لگانا

پودا لگانے کا عمل بھی ایک الگورتھم کے تحت کیا جاتا ہے:

(i) مناسب جگہ کا انتخاب کرنا (ii) زمین میں گڑھا کھودنا

stepacademyofficial.com

https://stepacademyofficial.com/

(iii) پودے کو صحیح طریقے سے رکھنا   (iv) مٹی ڈال کر دبانا

(v) پانی دینا اور ملچ ڈالنا   (vi) وقتاً فوقتاً پانی دینا تاکہ پودا نشوونما پائے

یہ تمام مراحل پودے کی اچھی نشوونما کے لیے ضروری ہیں۔ اگر ان میں سے کوئی مرحلہ چھوڑ دیا جائے، تو پودا صحیح طریقے سے نہیں بڑھے گا۔

الگورتھم کی اہمیت:

الگورتھم درج ذیل وجوہات کی بناپر اہم ہے:

(1) یہ مسائل کو منظم طریقے سے حل کرنے میں مدد دیتا ہے۔   (2) اس سے ہر مرحلہ درست ترتیب میں مکمل ہوتا ہے۔

(3) یہ عمل کو آسان اور قابلِ فہم بناتا ہے۔

(4) روزمرہ کے کاموں، جیسے راستہ تلاش کرنا، کھیل کھیلنا، یا اسکول کا بیگ پیک کرنا، میں مدد دیتا ہے۔

الگورتھم صرف کمپیوٹر میں ہی نہیں بلکہ ہماری روزمرہ زندگی میں بھی استعمال ہوتے ہیں، جو مسائل کو منطقی اور مؤثر طریقے سے حل کرنے میں مدد دیتے ہیں۔

7.2 کمپیوٹیشنل تھنکنگ کے اصول

سوال 5: کمپیوٹیشنل تھنکنگ میں 'مسئلے کو سمجھنے' کا تصور کیا ہے؟ مسئلے کو سمجھنا مسئلہ حل کرنے میں کیوں اہم ہے؟ مثالوں سے وضاحت کریں۔

جواب: مسئلے کو سمجھنا ایک ایسا عمل ہے جس میں کسی بھی مسئلے کی بنیادی نوعیت کو پہچانا جاتا ہے، اس کے تقاضے متعین کیے جاتے ہیں، اور اہداف طے کیے جاتے ہیں۔ یہ مسئلہ حل کرنے کا سب سے پہلا اور سب سے اہم مرحلہ ہے، خاص طور پر کمپیوٹیشنل تھنکنگ میں۔ جب کوئی شخص مسئلے کو مکمل طور پر سمجھ لیتا ہے تو وہ اس کے اہم اجزاء اور ضروریات کا تجزیہ کرسکتا ہے، جس سے حل کی طرف ایک منظم طریقہ اپنانا ممکن ہوتا ہے۔

مسئلے کو سمجھنے کی اہمیت:

-1 **وضاحت اور ارتکاز:** صحیح طریقے سے مسئلے کو سمجھنے سے یہ معلوم ہوتا ہے کہ اصل میں کیا حل کرنا ہے، جس سے غیر ضروری تفصیلات سے بچا جاسکتا ہے۔

-2 **اہداف کا تعین:** مسئلے کی واضح سمجھ سے ممکنہ اور قابلِ حصول اہداف طے کرنے میں مدد ملتی ہے۔

-3 **مؤثر حل:** جب مسئلے کو اچھی طرح سمجھا جائے تو اسے کم وسائل اور بہتر طریقوں سے حل کرنا آسان ہوجاتا ہے۔

-4 **غلطیوں سے بچاؤ:** اگر مسئلے کو غلط سمجھ لیا جائے تو حل بھی غلط ہوسکتا ہے، جس سے وقت اور محنت ضائع ہوسکتی ہے۔

مثال: اسکول کی ویب سائٹ بنانا

اگر کسی طالب علم کو اسکول کی ویب سائٹ بنانے کا کام دیا جائے، تو اسے کوڈنگ شروع کرنے سے پہلے مسئلے کو سمجھنا ہوگا، جس میں درج ذیل اقدامات شامل ہیں:

-1 **ضروریات کی شناخت:** ویب سائٹ میں کون سے فیچرز شامل ہونے چاہئیں، جیسے خبریں، تقریبات، کلاس شیڈول، اور رابطے کی تفصیلات۔

-2 **صارفین کی ضروریات:** یہ سمجھنا کہ ویب سائٹ کو طلبہ، اساتذہ یا والدین استعمال کریں گے، تاکہ اسے صارف دوست بنایا جاسکے۔

-3 **تکنیکی حدود:** ویب سائٹ بنانے کے لیے دستیاب وسائل، سرور تک رسائی، اور مطلوبہ سافٹ ویئر کا جائزہ لینا۔

اگر یہ تمام پہلوؤں کو اچھی طرح سمجھا جائے تو ویب سائٹ زیادہ مؤثر اور اسکول کی ضروریات کے مطابق بنائی جاسکتی ہے۔ لہٰذا، کمپیوٹیشنل تھنکنگ میں مسئلے کو سمجھنا نہایت ضروری ہے کیونکہ یہ بہتر حل اور مؤثر نتائج فراہم کرتا ہے۔

7.3 الگورتھم ڈیزائن کرنے کے طریقے

7.4 الگورتھم کی سرگرمیاں

سوال 6: فلو چارٹ کیا ہے؟ اس کی تفصیل سے وضاحت کریں۔

جواب: فلو چارٹ کسی عمل یا نظام کے مختلف مراحل کو تصویری شکل میں ظاہر کرنے کا ایک طریقہ ہے۔ اس میں مختلف علامات (Symbols) اور

stepacademyofficial.com

تیروں (Arrows) کا استعمال کیا جاتا ہے تاکہ مرحلہ وار عمل کو واضح طور پر دکھایا جا سکے۔ فلو چارٹ کا استعمال کمپیوٹر سائنس، انجینئرنگ، اور کاروبار کے مختلف شعبوں میں کیا جاتا ہے تاکہ عمل کو بہتر طور پر سمجھنے، منصوبہ بندی، اور بات چیت کے لیے استعمال کیا جا سکے۔

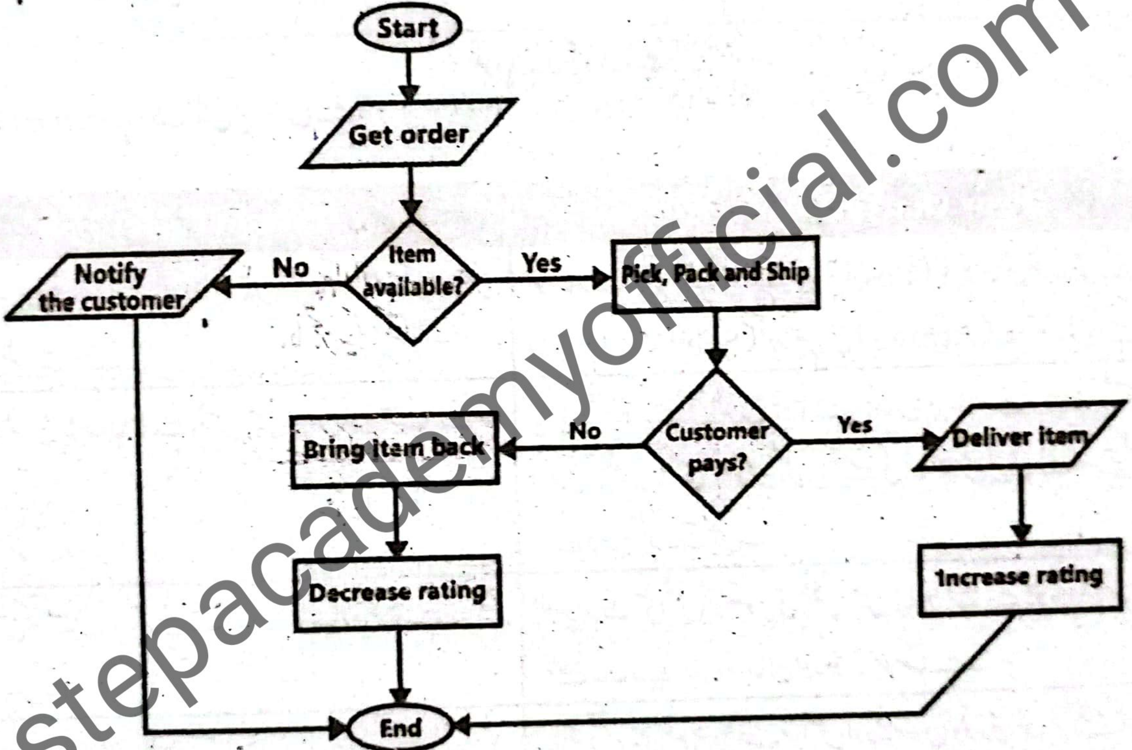
فلو چارٹ کی اہمیت:

1- **وضاحت:** فلو چارٹ کسی بھی عمل کو صاف اور آسان طریقے سے ظاہر کرتا ہے، جس سے اسے سمجھنا آسان ہو جاتا ہے۔

2- **ابلاغ (Communication):** یہ ایک ایسا آلہ ہے جو پیچیدہ عمل کو سمجھانے میں مدد دیتا ہے، تاکہ سب کو ایک ہی فہم حاصل ہو۔

3- **مسئلہ حل کرنے میں مدد:** فلو چارٹ سے کسی بھی عمل میں کمزوریاں یا رکاوٹیں آسانی سے دیکھی جا سکتی ہیں، جس سے بہتری کے لیے اقدامات کیے جا سکتے ہیں۔

4- **دستاویزی ریکارڈ:** فلو چارٹ کسی بھی نظام یا عمل کی تفصیلات کا ریکارڈ فراہم کرتا ہے، جو تربیت اور حوالہ کے لیے مفید ہوتا ہے۔



دکان کے آرڈر کے عمل کا فلو چارٹ

stepacademyofficial.com

سوال 7: کسی عمل کی نمائندگی کے لیے فلو چارٹ میں استعمال ہونے والی مختلف علامات (Symbols) کی وضاحت کریں۔

جواب: فلو چارٹ میں مختلف قسم کی علامات استعمال کی جاتی ہیں تاکہ عمل کو بہتر طریقے سے سمجھایا جا سکے۔ چند عام علامات درج ذیل ہیں:

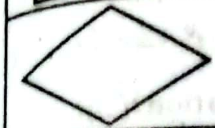
1- **اوول (Oval) شروع یا اختتام:** کسی عمل کی شروعات یا اختتام کو ظاہر کرنے کے لیے استعمال ہوتا ہے۔

2- **مستطیل (Rectangle) عمل:** کسی خاص کام یا عمل کی نشاندہی کرتا ہے، جیسے کوئی حسابی عمل یا ہدایت۔

3- **ڈائمنڈ (Diamond) فیصلہ:** کسی فیصلے (True/False یا Yes/No) کو ظاہر کرتا ہے، جس سے عمل دو مختلف سمتوں میں جا سکتا ہے۔

4- **تیر (Arrow) بہاؤ کی سمت:** فلو چارٹ میں عمل کی ترتیب یا سمت کو دکھانے کے لیے استعمال ہوتا ہے۔

استعمال بیان	نام	علامات
یہ کسی فلو چارٹ میں مرحلے کے بہاؤ (Flow) کا تعین کرنے کے لیے استعمال کیا جاتا ہے۔	فلو لائن (Flow Line)	→

یہ فلو چارٹ کے آغاز اور اختتام کی طرف اشارہ کرتا ہے۔	ٹرمینل (Terminal)	
یہ مقدار (Value) کے تبدیل ہونے کے آپریشن کی نمائندگی کرتا ہے۔	عمل (Processing)	
یہ ایک مشروط بیان ظاہر کرتا ہے جو اس بات کا تعین کرتا ہے کہ راستوں میں سے کون سا راستہ اختیار کیا جائے۔ آپریشن عام طور ایک ہاں/ نہیں کا سوال یا ایک صحیح/غلط ٹیسٹ ہے۔	فیصلہ سازی (Decission)	
یہ صارف ڈیٹا کے ان پٹ کے طور پر لینے کی نشاندہی کرتا ہے یا صارف کو نتائج دکھاتا ہے۔	ان پٹ/ آؤٹ پٹ (Input/Output)	
اگر ایک فلو چارٹ صفحے پر پورا نہیں آتا تب ہم ایک کنیکٹر (Connector) کے ذریعے فلو چارٹ کے حصوں کو ملا دیتے ہیں۔	کنیکٹر (Connector)	

فلو چارٹ کی علامات

سوال 8: سوڈو کوڈ اور فلو چارٹ میں کیا فرق ہے؟

جواب:

سوڈو کوڈ (Pseudo Code)	فلو چارٹ (Flow chart)
سوڈو کوڈ ایک سادہ زبان اور منظم انداز میں کسی الگورتھم کے مراحل بیان کرنے کا طریقہ ہے۔	فلو چارٹ کسی الگورتھم کے بہاؤ (Flow) کو ظاہر کرنے کے لیے گرافیکل علامات (Symbols) اور تیروں (Arrows) کا استعمال کرتا ہے۔
اسے کہانی کی طرح پڑھا جاتا ہے، جس میں ہر مرحلہ ترتیب سے لکھا جاتا ہے۔	یہ ایک فلم دیکھنے کی طرح ہوتا ہے، جہاں ہر علامت (جیسے مستطیل، ڈائمنڈ اور بیضوی شکلیں) کسی مخصوص عمل یا فیصلے کی نمائندگی کرتی ہے اور تیر عمل کے ربط اور سمت کو ظاہر کرتے ہیں۔
سوڈو کوڈ مراحل کو تفصیل اور کہانی کے انداز میں بیان کرتا ہے۔	فلو چارٹ کسی عمل کو تصویری شکل میں بیان کرتا ہے جو مجموعی بہاؤ اور ساخت کو سمجھنے کے لیے زیادہ آسان اور مؤثر ہو سکتا ہے۔
یہ الگورتھم کو اس انداز میں لکھنے کے لیے خاص طور پر مفید ہے کہ اسے آسانی سے کسی بھی پروگرامنگ زبان میں تبدیل کیا جا سکے۔	یہ الگورتھم میں موجود مختلف مراحل اور فیصلوں کو فوری طور پر سمجھنے میں مدد دیتا ہے۔

stepacademyofficial.com

سوال 9: آپ پیچیدہ کمپیوٹیشنل مسئلہ حل کرنے کے لیے الگورتھم ڈیزائن کے طریقے کس طرح استعمال کریں گے؟ اپنی وضاحت کو ایک تفصیلی مثال سے بیان کریں۔

جواب: الگورتھم ڈیزائن کے طریقے، جیسے فلو چارٹ اور سوڈو کوڈ، پیچیدہ کمپیوٹیشنل مسائل کو آسان اور منظم طریقے سے حل کرنے میں مدد دیتے ہیں۔ یہ طریقے مسئلے کے حل کی منطقی وضاحت فراہم کرتے ہیں اور اس کا نفاذ آسان بناتے ہیں۔

الگورتھم ڈیزائن کے طریقے استعمال کرنے کے اقدامات:

1- **مسئلے کی وضاحت:** مسئلے کو واضح طور پر بیان کریں، اس کے ان پٹ، آؤٹ پٹ اور ممکنہ حدود کو سمجھیں۔

2- **مسئلے کو چھوٹے حصوں میں تقسیم کرنا:** بڑے مسئلے کو چھوٹے، آسان حصوں میں تقسیم کریں تاکہ اسے سمجھنا اور حل کرنا آسان ہو۔

مثال: طلبہ کے نمبروں کی بنیاد پر گریڈ تفویض کرنے کے لیے الگورتھم بنائیں۔ گریڈنگ سسٹم درج ذیل ہے:

90 اور اس سے زیادہ: A+

https://stepacademyofficial.com/

A:89 سے 80

B:79 سے 70

C:69 سے 60

F:60 سے کم

الگورتھم: گریڈ اسائنمنٹ

ان پٹ: طالب علم کے حاصل کردہ نمبر پڑھیں (فرض کریں کہ اسے marks کہتے ہیں)

پروسیس: اگر 90≤marks ہو، تو گریڈ 'A+' ہوگا۔

اگر 80≤marks ہو، تو گریڈ 'A' ہوگا۔

اگر 70≤marks ہو، تو گریڈ 'B' ہوگا۔

اگر 60≤marks ہو، تو گریڈ 'C' ہوگا۔

اگر 60>marks ہو، تو گریڈ 'F' ہوگا۔

آؤٹ پٹ: اسائن شدہ گریڈ کو ظاہر کریں۔

سوال 10: آپ پیچیدہ کمپیوٹیشنل مسئلہ حل کرنے کے لیے فلو چارٹ اور سوڈوکوڈ جیسے الگورتھم ڈیزائن کے طریقے کس طرح استعمال کریں گے؟ اپنی وضاحت کو ایک تفصیلی مثال سے بیان کریں۔

جواب: الگورتھم ڈیزائن کے طریقے، جیسے فلو چارٹ اور سوڈوکوڈ، پیچیدہ کمپیوٹیشنل مسائل کو مرحلہ وار حل کرنے میں مدد دیتے ہیں۔ ان کے ذریعے مسئلے کی منطقی وضاحت ممکن ہوتی ہے اور اس کا نفاذ آسان ہو جاتا ہے۔

الگورتھم ڈیزائن کے طریقے استعمال کرنے کے اقدامات:

1- **مسئلے کی وضاحت:** مسئلے کو واضح طور پر بیان کریں اور اس کے ان پٹ، آؤٹ پٹ اور حدود کو سمجھیں۔

2- **مسئلے کو چھوٹے حصوں میں تقسیم کرنا:** مسئلے کو چھوٹے، زیادہ قابلِ فہم حصوں میں تقسیم کریں تاکہ اسے حل کرنا آسان ہو۔

3- **الگورتھم ڈیزائن کرنا:**

سوڈوکوڈ: سوڈوکوڈ لکھ کر الگورتھم کے مراحل کو ترتیب دیں۔ سوڈوکوڈ کسی خاص پروگرامنگ زبان پر منحصر نہیں ہوتا، اس لیے اسے لکھنا اور سمجھنا آسان ہوتا ہے۔

فلو چارٹ: فلو چارٹ بنائیں، جس میں مخصوص علامتیں استعمال کی جائیں (جیسے شروع/اختتام کے لیے بیضوی شکل، فیصلے کے لیے ہیرا اور پروسیس کے لیے مستطیل)۔

4- **جائزہ اور بہتری:** سوڈوکوڈ اور فلو چارٹ کو منطقی غلطیوں اور بہتری کے لیے پرکھیں۔

5- **الگورتھم کا نفاذ:** سوڈوکوڈ کو کسی پروگرامنگ زبان میں تبدیل کریں۔

6- **ٹیسٹنگ اور توثیق:** مختلف ٹیسٹ کیسز چلا کر تصدیق کریں کہ الگورتھم درست کام کر رہا ہے اور تمام ممکنہ صورتحال کا احاطہ کرتا ہے۔

مثال: دو عددوں کا سب سے بڑا مشترکہ مقسوم (GCD) تلاش کرنا

مسئلے کی وضاحت: ہم دو نمبروں کا GCD تلاش کرنا چاہتے ہیں۔ GCD وہ سب سے بڑی عددی قیمت ہے جو دونوں نمبروں کو بغیر باقی بچائے تقسیم کر سکتی ہے۔

مرحلہ 1: مسئلے کو چھوٹے حصوں میں تقسیم کرنا

یوکلڈین الگورتھم: اگر دوسرا عدد (b) صفر ہو، تو DCG پہلا عدد (a) ہوگا۔

stepacademyofficial.com

بصورت دیگر، GCD(a, b) = GCD(b, a % b) ہوگا۔

مرحلہ 2: الگورتھم ڈیزائن کرنا

سوڈوکوڈ:

```
FUNCTION GCD(a, b)
    WHILE b = 0 ? DO
    temp = b
    b = a MOD b
    a = temp
    END WHILE
    RETURN a
    END FUNCTION
```

فلوچارٹ:

1- شروع کریں
2- a اور b کو ان پٹ کریں
3- چیک کریں کہ b=0 ہے یا نہیں
ہاں: a=GCD اور اختتام کریں
نہیں: b کا نیا ویلیو a MOD b کے برابر رکھیں
4- a کو temp کے برابر کریں اور مرحلہ 3 پر واپس جائیں
5- GCD ظاہر کریں
6- اختتام کریں

نتیجہ: سوڈوکوڈ اور فلوچارٹ جیسے الگورتھم ڈیزائن کے طریقے کمپیوٹیشنل مسائل کو منظم طریقے سے حل کرنے میں مدد دیتے ہیں۔ ان طریقوں سے الگورتھم کی وضاحت آسان ہوتی ہے، جس سے اسے لاگو اور ڈیبگ کرنا آسان ہو جاتا ہے۔

سوال 11: فلوچارٹ اور سوڈوکوڈ کے درمیان موازنہ کریں اور ان کے فوائد و نقصانات بیان کریں۔ وضاحت کے لیے مثالیں بھی دیں۔

جواب: فلوچارٹ:

تعریف: فلوچارٹ کسی الگورتھم کی تصویری شکل ہوتی ہے، جس میں مختلف اشکال (جیسے اوول، مستطیل، ہیرا وغیرہ) کا استعمال کر کے کسی عمل کو واضح کیا جاتا ہے۔

فوائد:

بصری وضاحت: فلوچارٹ الگورتھم کو گرافک شکل میں دکھاتا ہے، جس سے عمل کو سمجھنا آسان ہو جاتا ہے۔

آسانی سے فالو کرنا: معیاری علامات کی مدد سے کوئی بھی شخص، چاہے وہ پروگرامنگ نہ جانتا ہو، آسانی سے سمجھ سکتا ہے۔

لوپس اور شرائط کی نشاندہی: فلوچارٹ کی مدد سے کسی الگورتھم میں موجود تکراری (loops) اور شرائط (conditions) کو واضح کیا جا سکتا ہے۔

نقصانات:

پیچیدگی: اگر الگورتھم بڑا ہو تو فلوچارٹ الجھن پیدا کر سکتا ہے۔

محدود اظہار: فلوچارٹ پروگرامنگ کی پیچیدہ تفصیلات کو مکمل طور پر نہیں دکھا سکتا۔

وقت طلب: تفصیلی فلوچارٹ بنانا خاص طور پر بڑے الگورتھمز کے لیے وقت طلب ہو سکتا ہے۔

stepacademyofficial.com

سوڈوکوڈ:

تعریف: سوڈوکوڈ کوڈ ایک تحریری شکل میں الگورتھم کی وضاحت کرتا ہے، جو پروگرامنگ زبان سے مشابہ ہوتا ہے مگر سادہ زبان میں لکھا جاتا ہے۔

فوائد:

سادگی: فلوچارٹ کے مقابلے میں سوڈوکوڈ لکھنا اور اس میں تبدیلی کرنا آسان ہوتا ہے۔

لچکدار: سوڈوکوڈ پیچیدہ منطق اور ڈیٹا اسٹرکچرز کو فلوچارٹ سے زیادہ واضح طریقے سے بیان کر سکتا ہے۔

زبان سے آزاد: سوڈوکوڈ کسی بھی مخصوص پروگرامنگ زبان پر منحصر نہیں ہوتا، اس لیے اسے بعد میں مختلف زبانوں میں تبدیل کرنا آسان ہوتا ہے۔

نتیجہ: فلوچارٹ اور سوڈوکوڈ دونوں اپنی جگہ اہم ہیں۔ فلوچارٹ تصویری طور پر الگورتھم کو سمجھنے میں مدد دیتا ہے، جبکہ سوڈوکوڈ الگورتھم کی تفصیلی وضاحت کے لیے موزوں ہوتا ہے۔ زیادہ تر مواقع پر، دونوں کو ایک ساتھ استعمال کیا جا سکتا ہے تاکہ الگورتھم کو بہتر طور پر سمجھا جا سکے۔

7.5 ڈرائی رن

سوال 12: فلوچارٹ کو 'ڈرائی رن' کرنے کے عمل کی وضاحت کریں اور دو عدد کے مجموعے کا حساب لگانے کی مثال سے اس کی وضاحت کریں۔

جواب: ڈرائی رن کسی فلوچارٹ یا الگورتھم کو بغیر کمپیوٹر کے دستی طور پر چیک کرنے کا عمل ہوتا ہے۔ اس کا مقصد یہ دیکھنا ہوتا ہے کہ الگورتھم منطقی طور پر درست کام کر رہا ہے یا نہیں۔

مثال: دو عدد کا مجموعہ

ہم ایک سادہ فلوچارٹ کا ڈرائی رن کرتے ہیں جو دو عدد کا مجموعہ نکالتا ہے:

1- شروع کریں
2- پہلا عدد درج کریں (فرض کریں: 3)
3- دوسرا عدد درج کریں (فرض کریں: 5)
4- مجموعہ = پہلا عدد + دوسرا عدد (3+5=8)
5- مجموعہ کو دکھائیں (نتیجہ: 8)
6- اختتام

ڈرائی رن کا مقصد:

فلوچارٹ کے ہر مرحلے کو جانچ کر غلطیوں کا پتہ لگایا جا سکتا ہے۔

الگورتھم کے منطقی بہاؤ کو سمجھنا آسان ہو جاتا ہے۔

کمپیوٹر پر چلانے سے پہلے مسائل کا پتہ لگایا جا سکتا ہے، جس سے ڈیبگنگ آسان ہو جاتی ہے۔

سوال 13: الگورتھمز میں 'ڈرائی رن' کے تصور کی وضاحت کریں اور اس کی اہمیت بیان کریں۔ فراہم کردہ متن اور خاکوں کی مدد سے مثال دیں۔

جواب: ڈرائی رن ایک ایسا طریقہ ہے جس میں الگورتھم کو دستی طور پر چیک کیا جاتا ہے تاکہ اس میں کسی قسم کی غلطی یا منطقی خامی کا پتہ لگایا جا سکے۔

ڈرائی رن کی اہمیت:

(1) یہ الگورتھم کی درستگی کو یقینی بناتا ہے۔
(2) یہ کمپیوٹر پر چلانے سے پہلے غلطیوں کو پکڑنے میں مدد دیتا ہے۔
(3) پیچیدہ الگورتھمز کی کارکردگی کو بہتر بنانے میں مدد ملتی ہے۔

مثال: فلوچارٹ کا ڈرائی رن

اگر ہمیں دو عدد کا مجموعہ نکالنا ہو، تو ہم دستی طور پر ہر قدم پر نظر ڈال سکتے ہیں اور دیکھ سکتے ہیں کہ نتیجہ درست ہے یا نہیں۔

مثال: سوڈوکوڈ کا ڈرائی رن

سوڈوکوڈ کو ہاتھ سے چیک کر کے دیکھا جا سکتا ہے کہ ہر لائن درست طریقے سے کام کر رہی ہے یا نہیں۔

نتیجہ: ڈرائی رن الگورتھمز کی تصدیق اور خرابیوں کی نشاندہی کے لیے ایک ضروری عمل ہے، جو کمپیوٹر پر کوڈ لکھنے سے پہلے مسائل کو حل کرنے میں مدد دیتا ہے۔

stepacademyofficial.com

سوال 14: سیمولیشن کیا ہے؟اس کی اہمیت، فوائد اور مثالیں تفصیل سے بیان کریں۔

جواب: سیمولیشن کا مطلب ہے کمپیوٹر پروگرامز کے ذریعے حقیقی دنیا کے کسی عمل یا نظام کا ماڈل بنانا۔ اس کی مدد سے ہم کسی چیز کو عملی طور پر آزمائے بغیر مختلف خیالات یا طریقے (الگورتھم) جانچ سکتے ہیں۔

سیمولیشن کی اہمیت:

1- **الگورتھم کی جانچ:** سیمولیشن سے الگورتھم کی کارکردگی کو مختلف ڈیٹا پر چیک کیا جا سکتا ہے۔ مثلاً، اگر ہمیں نمبروں کو ترتیب دینے کے ایک نئے طریقے کو آزمانا ہو تو ہم مختلف نمبروں کے سیٹ پر سیمولیشن کر کے اس کی رفتار دیکھ سکتے ہیں۔

2- **مختلف حالات کا مشاہدہ:** سیمولیشن کے ذریعے مختلف حالات پیدا کر کے ان کے نتائج دیکھے جا سکتے ہیں۔ مثلاً، اگر ہم پودوں کی نشوونما پر تحقیق کر رہے ہیں، تو ہم سیمولیشن کے ذریعے پانی اور دھوپ کی مختلف مقداریں آزما کر بہترین حالات معلوم کر سکتے ہیں۔

سیمولیشن کے فوائد:

1- **کم خرچ:** سیمولیشن اصل تجربات کے مقابلے میں کم لاگت اور تیزی سے کی جا سکتی ہے۔

2- **محفوظ:** خطرناک تجربات، جیسے عمارت میں آگ لگنے کی صورت حال، کو بغیر کسی جانی نقصان کے جانچا جا سکتا ہے۔

3- **بار بار دہرایا جا سکتا ہے:** سیمولیشن کو مختلف حالات میں کئی بار دہرایا جا سکتا ہے تاکہ نتائج کو بہتر سمجھا جا سکے۔

سیمولیشن کی مثالیں:

1- **موسم کی پیش گوئی:** ماہرین موسمیات کمپیوٹر ماڈلز میں درجہ حرارت، نمی اور ہوا کی رفتار کے ڈیٹا کو شامل کر کے موسم کی پیش گوئی کرتے ہیں۔

2- **ٹریفک نظام:** شہر کے منصوبہ ساز ٹریفک کے بہاؤ کا تجزیہ کرنے کے لیے سیمولیشن کا استعمال کرتے ہیں تاکہ سڑکوں اور اشاروں کی بہتری کے لیے منصوبہ بندی کی جا سکے۔

7.6 لاجک آف الگورتھمز فار ریزولیوشن آف پرابلمز (LARP) کا تعارف

سوال 15: LARP (Logic of Algorithms for Resolution of Problems) کی اہمیت بیان کریں اور یہ الگورتھم بنانے اور فلو چارٹس ڈیزائن کرنے میں کیسے مدد دیتا ہے؟

جواب: LARP ایک انٹرایکٹو طریقہ ہے جو طلبہ کو الگورتھم کے کام کرنے کے طریقے سیکھنے میں مدد دیتا ہے۔ اس میں طلبہ خود الگورتھم چلا کر دیکھ سکتے ہیں کہ ڈیٹا کس طرح پروسیس ہوتا ہے۔

LARP کی اہمیت:

1- **الگورتھم کو سمجھنے میں مدد:** مثلاً، اگر ہمیں کسی شخص کی سالانہ آمدنی پر ٹیکس لاگو ہونے کا پتہ لگانا ہو، تو LARP میں اس کا الگورتھم بنا کر جانچ سکتے ہیں۔

2- **مختلف ان پٹس کے اثرات دیکھنا:** اگر ان پٹ ویلیوز تبدیل کی جائیں تو آؤٹ پٹ پر کیا اثر پڑے گا؟ LARP اس کو واضح کرنے میں مدد دیتا ہے۔

3- **الگورتھم لکھنے اور بہتر بنانے کا موقع:** طلبہ خود الگورتھم لکھ سکتے ہیں، جس سے ان کی مسئلہ حل کرنے کی صلاحیت اور پروگرامنگ کی مہارت میں بہتری آتی ہے۔

LARP میں الگورتھم لکھنا: LARP میں الگورتھم ایک خاص ترتیب سے لکھے جاتے ہیں:

START سے آغاز ہوتا ہے۔

END پر ختم ہوتا ہے۔

WRITE کمانڈ پیغام دکھانے کے لیے

READ کمانڈ ڈیٹا لینے کے لیے

stepacademyofficial.com
https://stepacademyofficial.com/

IF...THEN...ELSE فیصلے کرنے کے لیے استعمال ہوتے ہیں۔

یہ طریقہ پیچیدہ مسائل کو چھوٹے حصوں میں تقسیم کر کے آسانی سے حل کرنے میں مدد دیتا ہے۔

LARP میں فلو چارٹس بنانا:

LARP میں فلو چارٹس مختلف علامتوں سے بنائے جاتے ہیں، جیسے:

مستطیل (Rectangle) عمل کے لیے

ہیرا (Diamond) فیصلوں کے لیے

متوازی چوکور (Parallelogram) ان پٹ اور آؤٹ پٹ کے لیے۔

یہ فلو چارٹس الگورتھم کے بہاؤ کو آسانی سے سمجھنے میں مدد دیتے ہیں۔

سوال 16: LARP کمپیوٹیشنل تھنکنگ کو کیسے بہتر بناتا ہے؟ اور ایک ایسی صورت حال بیان کریں جہاں LARP کسی الگورتھم کو بہتر بنانے میں مدد دے سکتا ہے؟

جواب: LARP کمپیوٹیشنل تھنکنگ کو بہتر بناتا ہے کیونکہ یہ طلبہ کو عملی طور پر مسئلے حل کرنے کا موقع فراہم کرتا ہے۔ اس کے ذریعے وہ بڑے مسائل کو چھوٹے حصوں میں تقسیم کر سکتے ہیں، پیٹرن پہچان سکتے ہیں، غیر ضروری معلومات ہٹا سکتے ہیں، اور مؤثر الگورتھمز بنا سکتے ہیں۔

LARP کے فوائد:

1- **عملی تجربہ:** طلبہ خود الگورتھم بنا کر اور اس کے نتائج دیکھ کر سیکھتے ہیں۔

2- **فوری فیڈ بیک:** جب الگورتھم غلط ہو تو فوری طور پر اس کی اصلاح کی جا سکتی ہے۔

3- **اشتراکی ماحول:** طلبہ مل کر مسائل حل کر سکتے ہیں، جس سے ان کی تخلیقی صلاحیتوں میں اضافہ ہوتا ہے۔

LARP کی مثال:

فرض کریں کہ ہمیں کسی لائبریری میں کتابیں قد کے لحاظ سے ترتیب دینی ہیں۔

LARP میں طلبہ مختلف کردار ادا کر سکتے ہیں جیسے 'لائبریرین'، 'کتابیں'، اور 'مشاہدہ کرنے والے'۔

لائبریرین پہلے سورٹ ببل (Sort Bubble) جیسے آسان الگورتھم استعمال کرتا ہے۔

پھر مختلف حالات کا جائزہ لے کر الگورتھم کو مزید مؤثر بنایا جا سکتا ہے۔

7.7 غلطی کی شناخت اور ڈیبگنگ

سوال 17: الگورتھمز اور فلو چارٹس میں مختلف قسم کی غلطیاں کون سی ہیں؟ ہر قسم کی وضاحت مثالوں کے ساتھ کریں۔

جواب: جب ہم LARP میں الگورتھمز یا فلو چارٹس بناتے ہیں، تو بعض اوقات غلطیاں ہو جاتی ہیں، جنہیں غلطیاں (Errors) یا بگز (Bugs) کہا جاتا ہے۔ یہ غلطیاں الگورتھم کے درست طریقے سے کام نہ کرنے کی وجہ بنتی ہیں۔

غلطیوں کی اقسام:

1- **سینٹیکس (Syntax) غلطیاں:** یہ غلطی الگورتھم یا فلو چارٹ میں غلطی سے کوئی قدم چھوڑنے یا غلط سمبل استعمال کرنے سے پیدا ہوتی ہے۔

مثال: اگر ہم LARP میں WRITE کی بجائے WRIT لکھ دیں، تو پروگرام کام نہیں کرے گا۔

2- **رن ٹائم (Runtime) غلطیاں:** یہ غلطی الگورتھم کے چلنے کے دوران پیدا ہوتی ہے جب کوئی ایسا عمل کیا جائے جو ممکن نہ ہو۔

مثال: اگر ہم کسی نمبر کو 0 سے تقسیم کریں تو کمپیوٹر اسے چلا نہیں سکتا اور پروگرام کریش ہو سکتا ہے۔

stepacademyofficial.com

-3 منطقی (Logical) غلطیاں: یہ غلطی الگورتھم کی منطق میں ہوتی ہے جس کی وجہ سے غلط نتائج حاصل ہوتے ہیں۔

مثال: اگر ہم کسی امتحان میں پاس ہونے کے لیے شرط رکھیں 'اگر نمبر 40 سے زیادہ ہوں تو فیل' تو یہ غلط لاجک ہے کیونکہ 40 سے زیادہ نمبر لینے والا طالبعلم پاس ہونا چاہیے۔

نتیجہ: LARP میں ایررز کو پہچاننے اور انہیں درست کرنے کے عمل کو ڈی بگنگ (Debugging) کہا جاتا ہے۔ اگر ہم غلطیوں کو صحیح وقت پر پہچان لیں اور انہیں درست کریں تو ہمارا الگورتھم مؤثر اور درست کام کرے گا۔

انشائی طرز کنسیپچوئل (Conceptual) سوالات

سوال 1: کمپیوٹیشنل تھنکنگ میں 'ڈی کمپوزیشن' کا تصور کیا ہے، اور اسے پرندوں کا گھر بنانے کے کام میں کیسے لاگو کیا جاسکتا ہے؟

جواب: ڈی کمپوزیشن ایک ایسا طریقہ ہے جس میں ایک پیچیدہ مسئلے کو چھوٹے اور آسان حصوں میں تقسیم کیا جاتا ہے۔ یہ کمپیوٹیشنل تھنکنگ کا ایک لازمی جزو ہے، کیونکہ اس کے ذریعے مشکل کاموں کو چھوٹے حصوں میں بانٹ کر آسانی سے سمجھا اور حل کیا جاسکتا ہے۔

پرندوں کا گھر بنانے میں ڈی کمپوزیشن کا اطلاق:

اگر ہم پرندوں کے لیے ایک گھر بنانا چاہتے ہیں، تو اس کام کو مختلف مراحل میں تقسیم کر سکتے ہیں:

-1 ڈیزائن تیار کرنا: گھر کا سائز، شکل اور ڈیزائن طے کرنا، ایک خاکہ بنانا اور ضروری پیمائش کرنا۔

-2 مواد اکٹھا کرنا: لکڑی، کیل، رنگ، ہتھوڑا، آری وغیرہ جیسے ضروری اوزار اور سامان جمع کرنا۔

-3 لکڑی کاٹنا: لکڑی کو مطلوبہ سائز میں ناپ کر کاٹنا۔

-4 گھر بنانا: تمام ٹکڑوں کو جوڑ کر پرندوں کا گھر تیار کرنا۔

-5 رنگ و سجاوٹ: پرندوں کے گھر کو پینٹ کرنا اور خوبصورت بنانے کے لیے سجاوٹ کرنا۔

-6 تنصیب کرنا: پرندوں کے لیے موزوں جگہ پر گھر نصب کرنا تاکہ وہ آسانی سے اس میں آسکیں۔

اس طرح کسی بھی بڑے کام کو چھوٹے حصوں میں تقسیم کرنے سے ہر مرحلے پر بہتر توجہ دی جاسکتی ہے، جس سے پورا کام آسان اور منظم ہو جاتا ہے۔

سوال 2: الگورتھم کی نمائندگی میں 'سوڈوکوڈ' اور 'فلو چارٹ' میں کیا فرق ہے؟ ان کی اہمیت مثالوں کے ساتھ بیان کریں۔

جواب: سوڈوکوڈ اور فلو چارٹ دونوں الگورتھم کی وضاحت کے طریقے ہیں، مگر ان کا استعمال اور انداز مختلف ہوتا ہے۔

سوڈوکوڈ:

تعریف: سوڈوکوڈ ایک سادہ اور آسان زبان میں لکھا جانے والا الگورتھم ہوتا ہے جو کہانی کے انداز میں ترتیب دیا جاتا ہے۔

خصوصیات: یہ الگورتھم کے مراحل کو تفصیل سے بیان کرتا ہے۔

اسے کسی بھی پروگرامنگ زبان میں آسانی سے تبدیل کیا جاسکتا ہے۔

مثال: ایک سوڈوکوڈ جو صارف کے نام اور پاس ورڈ کی تصدیق کرتا ہے:

پروسیجر CheckCredentials (نام، پاس ورڈ)

اگر نام اور پاس ورڈ درست ہیں

تو "لاگ ان کامیاب" دکھائیں

ورنہ

"غلط یوزر نیم یا پاس ورڈ" دکھائیں

اختتام

stepacademyofficial.com

https://stepacademyofficial.com/

فلو چارٹ: فلو چارٹ تصویری علامتوں اور تیروں کا استعمال کر کے الگورتھم کو گرافیکل انداز میں ظاہر کرتا ہے۔

خصوصیات: فلو چارٹ کسی بھی عمل کو آسان اور واضح طور پر دکھاتا ہے۔

یہ الگورتھم کے ہر مرحلے کو بصری انداز میں سمجھنے میں مدد دیتا ہے۔

مثال: اگر ہم سوڈوکوڈ میں دی گئی مثال کا فلو چارٹ بنائیں تو اس میں مستطیل (عمل)، ہیرا (فیصلہ) اور تیر (بہاؤ) کا استعمال ہوگا تاکہ یوزر نیم اور پاس ورڈ کی تصدیق کا عمل دکھایا جا سکے۔

اہمیت: سوڈوکوڈ پروگرامرز کے لیے مفید ہوتا ہے کیونکہ یہ الگورتھم کے منطق کو واضح کرتا ہے۔

فلو چارٹ پروگرامرز کے لیے فائدہ مند ہوتا ہے کیونکہ یہ الگورتھم کے بہاؤ کو آسانی سے سمجھنے میں مدد دیتا ہے۔

سوال 3: کمپیوٹیشنل تھنکنگ میں ''سیمولیشن'' کیا ہے؟ اس کی اہمیت، فوائد اور ایپلی کیشنز کی وضاحت کریں۔

جواب: **سیمولیشن کی تعریف:** سیمولیشن ایک ایسا طریقہ ہے جس میں کمپیوٹر پروگراموں کے ذریعے حقیقی دنیا کے کسی عمل یا نظام کی نقل تیار کی جاتی ہے۔

اس کے ذریعے مختلف نظریات اور الگورتھمز کو بغیر عملی تجربے کے جانچا جا سکتا ہے۔

سیمولیشن کیوں استعمال کی جاتی ہے؟

1- **الگورتھمز کی جانچ:** مختلف ڈیٹا سیٹس پر الگورتھم کی کارکردگی کو جانچنے کے لیے سیمولیشن کا استعمال کیا جاتا ہے۔

مثال: کسی نئے سورٹنگ الگورتھم کی رفتار اور کارکردگی کا جائزہ لینے کے لیے سیمولیشن کی جا سکتی ہے۔

2- **مناظر (سیناریو) کی جانچ:** مختلف حالات کا مشاہدہ کرنے کے لیے سیمولیشن کا استعمال ہوتا ہے۔

مثال: پودوں کی نشوونما پر پانی اور دھوپ کے اثرات کا تجزیہ کرنے کے لیے سیمولیشن کی جا سکتی ہے۔

فوائد:

1- **کم لاگت:** اصلی تجربات کے مقابلے میں یہ سستا اور تیز ہوتا ہے۔

2- **محفوظ:** خطرناک تجربات (جیسے آگ لگنے کی صورتحال) کو بغیر کسی نقصان کے آزمایا جا سکتا ہے۔

3- **دہرایا جا سکتا ہے:** سیمولیشن کو مختلف حالات میں بار بار چلا کر بہتر تجزیہ کیا جا سکتا ہے۔

مثالیں:

1- **موسمی پیش گوئی:** ماہرین موسمیات درجہ حرارت، ہوا کی رفتار اور نمی کے ڈیٹا کا استعمال کر کے موسم کی پیشن گوئی کے لیے سیمولیشن چلاتے ہیں۔

2- **ٹریفک مینجمنٹ:** شہر کے منصوبہ ساز سڑکوں اور ٹریفک سگنلز کے اثرات جانچنے کے لیے سیمولیشن استعمال کرتے ہیں تاکہ رش کو کم کیا جا سکے۔

اہمیت: سیمولیشن کمپیوٹیشنل تھنکنگ کا ایک لازمی جزو ہے جو حقیقی دنیا کے مسائل کو کم خرچ، محفوظ اور مؤثر طریقے سے حل کرنے میں مدد دیتا ہے۔

سوال 4: (Logic of Algorithms for Resolution of Problems)LARP کیا ہے؟ اس کی اہمیت اور الگورتھمز لکھنے میں اس کے کردار کی وضاحت کریں۔

جواب: **LARP کا تعارف:** LARP کا مطلب 'مسائل کے حل کے لیے الگورتھم کی منطق' ہے۔ یہ ایک انٹرایکٹو طریقہ ہے جس کے ذریعے الگورتھمز کو چلایا اور ان کے نتائج کا مشاہدہ کیا جا سکتا ہے۔

LARP کی اہمیت:

1- **الگورتھم کی تفہیم:** یہ مختلف مسائل کو حل کرنے کے طریقوں کو سیکھنے میں مدد دیتا ہے۔

مثال: کسی شخص کی سالانہ آمدنی پر ٹیکس لاگو ہوتا ہے یا نہیں، اسے LARP کے ذریعے جانچا جا سکتا ہے۔

stepacademyofficial.com

2- **ان پٹ-آؤٹ پٹ تجزیہ:** مختلف ان پٹ ویلیوز کے اثرات کو جانچنے کے لیے استعمال کیا جاتا ہے۔

3- **الگورتھم لکھنے میں مدد:** LARP الگورتھمز کو ایک منظم طریقے سے لکھنے کی سہولت فراہم کرتا ہے۔

LARP کے ذریعے الگورتھم لکھنے کا طریقہ:

سادہ اور واضح جملے: ہر الگورتھم 'START' سے شروع ہوتا ہے اور 'END' پر ختم ہوتا ہے۔

بنیادی ہدایات: 'WRITE' (پیغام دکھانے کے لیے) اور 'READ' (صارف سے ان پٹ لینے کے لیے) استعمال کیے جاتے ہیں۔

فیصلہ سازی: 'IF...THEN...ELSE' کا استعمال کیا جاتا ہے تاکہ مختلف حالات میں مناسب فیصلے کیے جا سکیں۔

نتیجہ: LARP الگورتھم سیکھنے کا ایک طاقتور طریقہ ہے جو مسائل کو منطق کی بنیاد پر حل کرنے میں مدد دیتا ہے۔ اس کے ذریعے سیکھنے والے بنیادی پروگرامنگ تصورات کو آسانی سے سمجھ سکتے ہیں۔

دلچسپ معلومات

یہ پیچیدہ مسائل کو حل کرتے وقت انہیں چھوٹے حصوں میں توڑنے کی کوشش کریں اور اہم مراحل پر توجہ مرکوز کریں۔ یہ آپ کو مسئلے کو بہتر طور پر سمجھنے میں مدد دے گا اور حل تلاش کرنا زیادہ آسان بنائے گا۔ ابstraction کا استعمال کرتے ہوئے، ہم پیچیدہ مسائل کا سامنا کر سکتے ہیں۔

کسی مسئلے کو حل کرنا شروع کرنے سے پہلے ہمیشہ اسے اچھی طرح سے سمجھنے کے لیے وقت نکالیں۔ سوالات پوچھیں، معلومات جمع کریں اور کسی بھی شک کی وضاحت کریں۔ یہ بنیادی قدم بہتر اور زیادہ مؤثر حل کا باعث بنے گا۔

جب آپ ایک الگورتھم لکھ رہے ہوتے ہیں تو اس بات کو مدِ نظر رکھیں کہ وہ ٹاسک کتنے مراحل میں مکمل کرتا ہے۔ کم مراحل سے مراد تیز الگورتھم ہے۔

ہمیشہ غلطی کے پیغام کو غور سے پڑھیں یہ عموماً آپ کو بتا دیتے ہیں کہ مسئلہ کہاں پر ہے۔

کیا آپ جانتے ہیں؟

کمپیوٹیشنل تھنکنگ صرف کمپیوٹر سائنس تک محدود نہیں ہے۔ یہ روزمرہ کے مسائل کو حل کرنے میں استعمال ہوتی ہے، جیسے ایک سفر کی منصوبہ بندی کرنا یا کاموں کی ترتیب دینا۔

1 سے 10 تک سائیڈ لمبائی کے ساتھ ایک ٹیبل بنائیں۔ لگاتار طاق نمبروں کو جمع کرنے کے پیٹرن کا استعمال کرتے ہوئے مربعوں کے رقبوں کا حساب لگائیں۔ سائیڈ کی لمبائی کو ترتیب دے کر اپنے نتائج کی تصدیق کریں اور دیکھیں کہ آیا پیٹرن برقرار ہے یا نہیں۔

الگورتھم صرف کمپیوٹرز میں ہی استعمال نہیں ہوتے ہیں؟ وہ ہر جگہ موجود ہیں! جب آپ اپنے دوست کے گھر جانے کے لیے ہدایات پر عمل کرتے ہیں یا قواعد کے ساتھ بورڈ گیم کھیلتے ہیں تو آپ الگورتھم استعمال کر رہے ہوتے ہیں۔ الگورتھمز ہمیں منطقی طریقے سے مسائل کو حل کرنے میں مدد کرتے ہیں۔

فلو چارٹ کو کمپیوٹر سائنس دانوں جیسے کہ جان وان نیومین اور ہرمن گولڈسٹائن نے کمپیوٹنگ کے ابتدائی دنوں میں مقبول بنایا۔

اصل کوڈ لکھنے سے پہلے سافٹ ویئر کی تیاری اور سوڈو کوڈ کو اکثر استعمال کیا جاتا ہے تاکہ اس بات کو یقینی بنایا جا سکے کہ منطق درست ہے یہ ٹیم کے افراد کے مابین بات چیت کو آسان بناتا ہے خاص طور پر جب وہ رابطے مختلف پروگرامنگ لینگویجز استعمال کر رہے ہیں۔

الگورتھم کی کمپلیکسٹی کو Big O نوٹیشن کے ذریعے ظاہر کی جاتی ہے، جیسے O(n)، O(logn) یا O(n2)۔ یہ ہمیں مختلف الگورتھمز کا موازنہ کرنے میں مدد دیتی ہے [illegible]

stepacademyofficial.com

کچھ الگورتھم ایک ہی کام کو دوسروں کے مقابلے میں بہت تیزی سے انجام دے سکتے ہیں۔ مثال کے طور پر، 100 اشیاء کی فہرست کو ترتیب دینے میں ایک الگورتھم 1 سیکنڈ جب کہ دوسرا الگورتھم 10 سیکنڈ کا وقت لے سکتا ہے!

اپنے کوڈ یا الگورتھم کی ڈرائی رننگ کرنے سے ڈویلپمنٹ کے عمل میں غلطیوں کو جلد پکڑنے میں مدد ملتی ہے، جس سے وقت اور محنت کی بچت ہوتی ہے۔
بہت سے پیشہ ور پروگرامرز اور کمپیوٹر سائنسدان اپنے الگورتھم کو صحیح طریقے سے کام کرنے کو یقینی بنانے کے لیے ڈرائی رننگ کوڈ۔ ہیکنگ تکنیک کے طور پر استعمال کرتے ہیں!

سنٹیکس کی غلطیوں کو تلاش کرنا سب سے آسان ہے کیونکہ LARP ٹول عام طور پر ان کی نشان دہی کرتا ہے۔ تاہم منطقی غلطیوں کو تلاش کرنا سب سے مشکل ہے۔ کیونکہ الگورتھم اب بھی چلتا ہے لیکن صحیح جوابات نہیں دیتا۔

''ڈی بگنگ'' کی اصطلاح ایک حقیقی بگ یعنی کیڑے (Moth) سے آئی ہے جو ابتدائی کمپیوٹر میں مسائل پیدا کرتا تھا۔ جب اس کیڑے کو ہٹایا گیا تو اس عمل کو ''ڈی بگنگ'' کہا گیا۔

معروضی سوالات

محکمہ تعلیم کی نئی امتحانی تکنیکس (Knowledge, Understanding, Application, Analytical & Conceptual) کی روشنی میں مرتب کیے گئے کثیر الانتخابی سوالات

7.1 کمپیوٹیشنل تھنکنگ کی تعریف

☆ درست جواب کے گرد دائرہ لگائیں۔

-1 کمپیوٹیشنل تھنکنگ بنیادی طور پر کیا ہے؟
(A) ایک پروگرامنگ زبان
(B) مسئلہ حل کرنے کا ایک طریقہ
(C) کمپیوٹر کا ایک ہارڈ ویئر حصہ
(D) ڈیٹا ذخیرہ کرنے کی ایک تکنیک

-2 کمپیوٹیشنل تھنکنگ کے بارے میں کون سا بیان درست ہے؟
(A) یہ صرف کمپیوٹر سائنس میں استعمال ہوتی ہے۔
(B) یہ صرف پروگرامنگ کے کاموں تک محدود ہے۔
(C) اسے حیاتیات اور ریاضی جیسے مختلف شعبوں میں استعمال کیا جا سکتا ہے۔
(D) اسے روزمرہ کی زندگی میں استعمال نہیں کیا جا سکتا

-3 مسئلہ حل کرنے میں ڈی کمپوزیشن (Decomposition) کا مقصد کیا ہوتا ہے؟
(A) کسی مسئلے کو مزید پیچیدہ بنانا
(B) کسی مسئلے کو چھوٹے، قابلِ انتظام حصوں میں تقسیم کرنا
(C) بغیر کسی منصوبہ بندی کے مسئلہ حل کرنا
(D) کئی مسائل کو ایک ساتھ جوڑنا

-4 علی ایک درخت کے اوپر ایک جھونپڑی بنانا چاہتا ہے لیکن اسے یہ کام بہت مشکل لگ رہا ہے۔ وہ ڈی کمپوزیشن کا استعمال کیسے کر سکتا ہے؟
(A) بغیر کسی منصوبہ بندی کے پوری جھونپڑی ایک ساتھ بنا لے
(B) بغیر سوچے سمجھے سامان خرید کر جھونپڑی بنانا شروع کر دے
(C) منصوبہ بندی کرے اور کام کو چھوٹے حصوں میں تقسیم کرے جیسے ڈیزائننگ، سامان اکٹھا کرنا، لکڑی کاٹنا، جھونپڑی جوڑنا اور سجاوٹ کرنا
(D) انتظار کرے کہ کوئی آسان طریقہ مل جائے

stepacademyofficial.com

-5 ڈی کمپوزیشن کے اصول کے مطابق پرندوں کا گھر بنانے کا صحیح طریقہ کون سا ہے؟

(A) پرندوں کا گھر لگائیں → سامان اکٹھا کریں → رنگ وسجاوٹ کریں → لکڑی کاٹیں

(B) سامان اکٹھا کریں → لکڑی کاٹیں → ٹکڑوں کو جوڑیں → رنگ وسجاوٹ کریں → پرندوں کا گھر لگائیں؟

(C) پرندوں کا گھر ڈیزائن کریں → ٹکڑوں کو جوڑیں → لکڑی کاٹیں → پرندوں کا گھر لگائیں → رنگ وسجاوٹ کریں

(D) لکڑی کاٹیں → رنگ وسجاوٹ کریں → ٹکڑوں کو جوڑیں → پرندوں کا گھر ڈیزائن کریں → پرندوں کا گھر لگائیں

-6 پیٹرن کی شناخت (Recognition Pattern) کا مسئلہ حل کرنے میں کیا کردار ہوتا ہے؟

(A) فارمولے یاد کرنے میں مدد دیتا ہے، چاہے ان کا مطلب نہ سمجھا جائے

(B) مسئلہ حل کرنے میں تیزی لانے کے لیے قواعد پہچاننے میں مدد دیتا ہے

(C) صرف کسی دیے گئے مسئلے میں غلطیاں تلاش کرنے پر توجہ دیتا ہے

(D) صرف ریاضی کے حساب کتاب میں استعمال ہوتا ہے

-7 اگر مربع کے رقبے کا پیٹرن جاری رہے، تو 8 لمبائی والے مربع کا رقبہ کیا ہوگا؟

(A) 49   (B) 56   (C) 64   (D) 72

-8 مسئلہ حل کرنے میں تجدید (Abstraction) کا بنیادی مقصد کیا ہے؟

(A) مزید تفصیلات شامل کر کے مسئلے کو پیچیدہ بنانا   (B) صرف اہم تفصیلات پر توجہ دے کر مسئلہ آسان بنانا

(C) مسئلے کی تمام تفصیلات مکمل طور پر ہٹا دینا   (D) مسئلے کو کسی اور بالکل مختلف مسئلے سے بدل دینا

-9 درج ذیل میں سے کون سی مثال تجدید (Abstraction) کی بہترین وضاحت کرتی ہے؟

(A) ایک باورچی پورا نسخہ پڑھنے کے بعد ہی کوئی کھانا پکائے

(B) ایک شخص چائے بنانے کے لیے ہدایت پر عمل کرے، چاہے اسے چائے کے کیمیکل اجزاء کا علم نہ ہو

(C) ایک پروگرامر کوڈ لکھتے ہوئے ہارڈ ویئر کی تمام تفصیلات شامل کرے

(D) ایک طالب علم پوری درسی کتاب رٹا لگائے بغیر بنیادی تصورات سمجھے

-10 تجدید (Abstraction) کے بارے میں کون سا بیان غلط ہے؟

(A) یہ غیر ضروری تفصیلات چھپا کر پیچیدگی کم کرنے میں مدد دیتی ہے

(B) یہ مسئلہ حل کرنے کے لیے صرف ضروری تفصیلات پر توجہ دیتی ہے

(C) یہ مسئلے کو مزید مشکل بنا دیتی ہے کیونکہ اس میں زیادہ تفصیلات شامل ہوتی ہیں

(D) یہ مسئلے کو ڈیزائن اور حل کرنے میں مددگار ثابت ہوتی ہے

-11 الگورتھم کی سب سے بہترین تعریف کیا ہے؟

(A) ایک کیک بنانے کے لیے استعمال ہونے والے اجزاء کی فہرست

(B) کسی بھی کام کو مکمل کرنے کے لیے بے ترتیب ہدایات کا مجموعہ

(C) مسئلہ حل کرنے کے لیے مرحلہ وار درست طریقہ کار   (D) پروگرامنگ میں استعمال ہونے والے نمبروں کا ایک سیٹ

-12 کیک پکانے کے الگورتھم میں کون سا مرحلہ شامل ہونا ضروری ہے تاکہ کیک درست طریقے سے بنے؟

(A) بیٹر کو اچھی طرح مکس کرنا اور پھر پین میں ڈالنا   (B) کیک پکنے کے بعد فوراً کھا لینا

(C) اوون کا درجہ حرارت زیادہ سے زیادہ رکھنا   (D) کیک پکنے کے بعد مزید اجزاء شامل کرنا

stepacademyofficial.com

13- درج ذیل میں سے کون سی سرگرمی الگورتھم کی مثال نہیں ہے؟

(A) کسی ترکیب پر عمل کرتے ہوئے کھانے کی تیاری (B) کسی پودے کو بغیر کسی خاص ترتیب کے پانی دینا

(C) کسی دوست کو راستہ سمجھانے کے لیے مرحلہ وار ہدایات دینا

(D) ریاضی کے سوال کو کسی مخصوص طریقہ سے حل کرنا

14- آئن سٹائن نے مسئلے کو سمجھنے میں زیادہ وقت لگانے پر زور کیوں دیا تھا؟

(A) کیونکہ مسئلہ حل کرنا ہمیشہ مسئلہ سمجھنے سے آسان ہوتا ہے (B) تاکہ مسئلہ حل کرنے کا ایک مؤثر اور منظم طریقہ اپنایا جا سکے

(C) کیونکہ مسائل خود ہی حل ہو جاتے ہیں بغیر زیادہ محنت کے (D) تاکہ غیر ضروری باتوں کو چھوڑ کر صرف کوڈنگ پر توجہ دی جائے

7.2 کمپیوٹیشنل تھنکنگ کے اصول

15- ایک طالبعلم اپنے اسکول کے لیے ویب سائٹ ڈیزائن کر رہا ہے۔ کوڈنگ شروع کرنے سے پہلے، وہ یہ تجزیہ کرتا ہے کہ کن خصوصیات کی ضرورت ہے، صارفین کون ہوں گے، اور کون سے تکنیکی وسائل دستیاب ہیں۔ وہ مسئلہ حل کرنے کے کس مرحلے پر عمل کر رہا ہے؟

(A) حل پر عملدرآمد (Implementation Solution)

(B) مسئلے کو سمجھنا (Understanding Problem)

(C) خرابیوں کی جانچ اور درستگی (Testingand Debugging)

(D) کوڈ کی بہتری (Optimization Code)

16- ایک مسئلے کو غلط سمجھنے سے محنت ضائع کیوں ہو سکتی ہے

(A) اگر مسئلہ صحیح نہ سمجھا جائے تو اس کا حل بھی درست نہیں ہوگا۔ (B) کوئی بھی حل مسئلے کو سمجھے بغیر بھی کام کر سکتا ہے۔

(C) مسئلہ حل کرنا صرف اندازے لگانے کا عمل ہے، اس لیے اسے سمجھنا ضروری نہیں ہے۔

(D) تمام حل ہمیشہ مکمل ہوتے ہیں، چاہے مسئلہ صحیح سمجھا گیا ہو یا نہیں۔

17- مسئلہ آسان بنانے (Problem Simplification) کا مقصد کیا ہے؟

(A) کسی پیچیدہ مسئلے کو چھوٹے اور قابلِ حل حصوں میں تقسیم کرنا۔ (B) سب سے آسان حل تلاش کرنا، چاہے وہ مؤثر نہ ہو۔

(C) غیر ضروری تفصیلات چھوڑ کر مسئلے کے اصل نکتہ پر توجہ دینا۔ (D) مسئلہ حل کرنے کے مراحل کو فلو چارٹ کے ذریعے ظاہر کرنا۔

18- مسائل کو حل کرنے کے لیے مختلف الگورتھم ڈیزائن طریقے اپنانے کا سب سے بڑا فائدہ کیا ہے؟

(A) ہر مسئلے کو سب سے زیادہ موثر طریقے سے حل کرنا یقینی بناتا ہے۔

(B) مسئلے کی نوعیت کے مطابق موزوں ترین طریقہ منتخب کرنے میں مدد دیتا ہے۔

(C) پروگرامرز کو مختلف طریقے استعمال کر کے اپنی مہارت دکھانے کا موقع ملتا ہے۔

(D) اس سے مسئلے کو آسان بنانے کی ضرورت ختم ہو جاتی ہے، کیونکہ ہر طریقہ ہر مسئلے پر کام کرتا ہے۔

7.3 الگورتھم ڈیزائن کرنے کے طریقے

19- فلو چارٹس مسئلہ حل کرنے اور اصلاح میں کس طرح مدد دیتے ہیں؟

(A) یہ حل کو بصری (Visual) طور پر ظاہر کرتے ہیں، جس سے اسے سمجھنا آسان ہو جاتا ہے۔

(B) یہ عمل میں موجود رکاوٹوں اور کمزوریوں کی نشاندہی کرتے ہیں۔

stepacademyofficial.com

(C) یہ دوسروں کے ساتھ حل شیئر کرنے کے لیے ایک مواصلاتی ذریعہ فراہم کرتے ہیں۔

(D) ان سب کے ذریعے

20- مؤثر مسئلہ حل کرنے کے لیے مختلف الگورتھم ڈیزائن طریقوں کو سمجھنا کیوں ضروری ہے؟

(A) تاکہ پرانے طریقوں سے بچا جا سکے۔ (B) تاکہ نوکری حاصل کرنے کے لیے زیادہ علم کا مظاہرہ کیا جا سکے۔

(C) تاکہ مسئلے کی نوعیت کے مطابق سب سے موزوں طریقہ اپنایا جا سکے۔

(D) تاکہ ہر مسئلے کو سب سے زیادہ پیچیدہ طریقے سے حل کیا جا سکے۔

21- فلوچارٹ کے حوالے سے درج ذیل میں سے کون سا بیان درست ہے؟

(A) فلوچارٹس صرف پیچیدہ مسائل کے لیے مفید ہوتے ہیں۔

(B) فلوچارٹس بنیادی طور پر مواصلات اور دستاویزات کے لیے استعمال ہوتے ہیں، مسئلہ حل کرنے کے لیے نہیں۔

(C) فلوچارٹس کسی عمل میں رکاوٹوں اور کمزوریوں کی نشاندہی کرنے میں مدد دیتے ہیں، جس سے بہتری ممکن ہوتی ہے۔

(D) فلوچارٹ ایک قسم کی پروگرامنگ زبان ہے۔

7.6 لاجک آف الگورتھمز فار ریزولیوشن آف پرابلمز (LARP) کا تعارف

stepacademyofficial.com

22- LARP کو الگورتھم سیکھنے کا ایک انٹرایکٹو (Interactive) طریقہ کیوں سمجھا جاتا ہے؟

(A) یہ صرف نظریاتی تصورات پر توجہ دیتا ہے۔ (B) یہ طلبہ کو الگورتھم یاد کرنے میں مدد دیتا ہے۔

(C) یہ الگورتھمز پر تجربات کرنے اور ان کے نتائج دیکھنے کے قابل بناتا ہے۔

(D) یہ الگورتھمز لکھنے کی ضرورت کو ختم کر دیتا ہے۔

23- LARP میں الگورتھمز لکھنے کے بارے میں کون سا بیان درست ہے؟

(A) LARP میں الگورتھمز کے لیے START اور END کمانڈز کی ضرورت نہیں ہوتی۔

(B) LARP پیچیدہ کوڈنگ سینٹیکس (Syntax) استعمال کرتا ہے۔

(C) LARP منطقی بہاؤ (Logical Flow) پر توجہ دیتا ہے، بغیر پیچیدہ کوڈنگ کے۔

(D) LARP میں فیصلے (Decision-Making) کی سہولت نہیں ہوتی۔

24- LARP میں صارف کو پیغامات ظاہر کرنے کے لیے کون سی کمانڈ استعمال کی جاتی ہے؟

(A) READ (B) WRITE (C) INPUT (D) START

25- اگر دیے گئے LARP الگورتھم میں ان پٹ نمبر 7 ہو، تو آؤٹ پٹ کیا ہوگا؟

```
START
WRITE "Enter a number"
READ number
IF number % 2 == 0 THEN
WRITE "The number is even"
ELSE
WRITE "The number is odd"
ENDIF
END
```

(A) The number is even (B) The number is odd

(C) الگورتھم میں خرابی ہے   (D) کوئی آؤٹ پٹ نہیں ہوگا

-26 **الگورتھم میں سب سے مشکل پہچانے جانے والی خرابی کون سی ہوتی ہے؟**

(A) Syntax Error   (B) Run-time Error   (C) Logical Error   (D) Compilation Error

-27 **اگر ایک طالبعلم الگورتھم میں خرابی تلاش کر رہا ہو لیکن وہ یہ نہیں سمجھ پا رہا کہ غلطی کہاں ہے، تو ڈیبگنگ (Debugging) کے عام طریقہ کار کے مطابق پہلا قدم کیا ہونا چاہیے؟**

(A) تبصرے (Comments) کا استعمال   (B) مراحل کا سراغ لگانا (Trace the Steps)

(C) شرائط کو چیک کرنا (Check Conditions)   (D) مسئلے کو آسان بنانا (Simplify the Problem)

-28 **اگر ایک طالبعلم LARP الگورتھم میں 'Undefined Variable' کی خرابی کا سامنا کرتا ہے، تو ممکنہ وجہ کیا ہو سکتی ہے؟**

(A) متغیر (Variable) کو استعمال کرنے سے پہلے اسے ڈیفائن (Define) نہ کرنا

(B) الگورتھم میں کوئی قدم بھول جانا

(C) فیصلے (Decision Step) میں غلط شرط (Condition) استعمال کرنا

(D) غلط ریاضیاتی عمل انجام دینا۔

-29 **LARP میں کون سا منظر نامہ زیادہ امکان رکھتا ہے کہ رَن ٹائم ایرر (Run-Time Error) پیدا کرے؟**

(A) الگورتھم میں غلط علامت (Symbol) استعمال کرنا   (B) فیصلے (Decision Step) میں غلط شرط لکھنا

(C) چلانے کے دوران کسی عدد کو صفر سے تقسیم کرنے کی کوشش کرنا

(D) کوڈ میں تبصرے (Comments) شامل کرنا بھول جانا

-30 **منطقی غلطیاں (Logical Errors) الگورتھم کو چلنے سے نہیں روکتیں، لیکن غلط نتائج پیدا کرتی ہیں، اس کی کیا وجہ ہے؟**

(A) کیونکہ یہ عملدرآمد (Execution) سے پہلے پہچان لی جاتی ہیں۔

(B) کیونکہ یہ منطق پر اثر ڈالتی ہیں، لیکن عملدرآمد کو نہیں روکتیں۔

(C) کیونکہ یہ خودکار طور پر شناخت اور درست کر لی جاتی ہیں۔   (D) کیونکہ یہ درحقیقت نحو (Syntax) کی غلطیاں ہوتی ہیں۔

جوابات

(C) -10	(B) -9	(B) -8	(C) -7	(B) -6	(B) -5	(C) -4	(B) -3	(C) -2	(B) -1
(C) -20	(D) -19	(B) -18	(A) -17	(A) -16	(B) -15	(B) -14	(B) -13	(A) -12	(C) -11
(B) -30	(C) -29	(A) -28	(B) -27	(C) -26	(B) -25	(B) -24	(C) -23	(C) -22	(C) -21

کثیر الانتخابی کنسیپچوئل (Conceptual) سوالات

☆ درست جواب کے گرد دائرہ لگائیں۔

-1 **الگورتھم کے تناظر میں مرحلہ وار طریقہ کیوں ضروری ہے؟**

(A) یہ ایک ساتھ کئی مسائل حل کرنے کی اجازت دیتا ہے۔

(B) یہ مسئلے کو بے ترتیب مراحل میں تقسیم کرنے کو یقینی بناتا ہے۔

(C) یہ کسی خاص مقصد کو مؤثر طریقے سے حاصل کرنے کے لیے ایک منظم طریقہ فراہم کرتا ہے۔

(D) یہ صرف نظریاتی تصورات پر توجہ دیتا ہے اور عملی اطلاق نہیں رکھتا۔

stepacademyofficial.com

-2 درخت لگانے کے عمل کو ایک الگورتھمک عمل کیوں سمجھا جاتا ہے؟

(A) اس میں غیر ترتیب شدہ ہدایات کا مجموعہ ہوتا ہے۔

(B) اس میں کسی مقصد کو حاصل کرنے کے لیے ترتیب وار مراحل شامل ہوتے ہیں۔

(C) یہ کسی منظم منصوبے کی بجائے صرف وجدان (intuition) پر انحصار کرتا ہے۔

(D) اس میں کوئی خاص نمونہ یا منطق شامل نہیں ہوتی۔

-3 ایک طالب علم کو دانت برش کرنے کا الگورتھم بنانے کا کام دیا گیا ہے۔ اس میں سب سے زیادہ اہم پہلو کیا ہونا چاہیے؟

(A) برش کرنے میں شامل ہر ایک حرکت کو فہرست میں شامل کرنا۔

(B) وہ بنیادی ترتیب وار مراحل جو مؤثر برشنگ کو یقینی بنائیں۔

(C) برش کرنے کی اہمیت کی پیچیدہ وضاحت دینا۔

(D) ترتیب کو نظر انداز کرنا اور عمومی ہدایات پر توجہ دینا۔

-4 ایک سافٹ ویئر ڈویلپر ایک خودکار نظام ڈیزائن کر رہا ہے جو صارفین کو آن لائن درخواست کے عمل میں رہنمائی فراہم کرے گا۔ دی گئی معلومات کے مطابق کون سا اصول سب سے زیادہ اہم ہوگا؟

(A) تفصیلی مرحلہ وار طریقہ کار سے گریز کیا جائے تاکہ عمل آسان رہے۔

(B) درخواست کو ایک منظم الگورتھم کے تحت واضح مراحل کے ساتھ بنایا جائے۔

(C) صارفین کو بغیر کسی پہلے سے طے شدہ مراحل کے مکمل آزادی دی جائے تاکہ لچک میں اضافہ ہو۔

(D) تمام تفصیلات کو ایک ساتھ ظاہر کیا جائے تاکہ وضاحت میں بہتری آئے۔

-5 طلباء کا ایک گروپ سینڈوچ بنانے کے الگورتھم کا تبادلہ کر رہا ہے۔ اس سرگرمی کی کامیابی کو یقینی بنانے میں سب سے اہم عنصر کیا ہے؟

(A) الگورتھم میں ہر مرحلے کی وضاحت اور اہمیت۔

(B) الگورتھم میں درج اجزاء کی تعداد۔

(C) ہر طالب علم کی ترجیح کے مطابق مراحل کو دوبارہ ترتیب دینے کی لچک۔

(D) کم سے کم وقت میں سینڈوچ مکمل کرنے کی صلاحیت۔

-6 ایک پیچیدہ مسئلے کو چھوٹے ذیلی مسائل میں تقسیم کرنے سے حل تلاش کرنے میں کس طرح مدد ملتی ہے؟

(A) یہ صرف ایک پہلو پر توجہ مرکوز کرنے کی اجازت دیتا ہے جبکہ دیگر کو نظر انداز کرتا۔

(B) یہ مسئلہ حل کرنے کے عمل کو آسان اور قابلِ انتظام بناتا ہے۔

(C) یہ حل کی پیچیدگی میں اضافہ کرتا ہے کیونکہ مزید تفصیلات شامل ہو جاتی ہیں۔

(D) یہ واضح مقاصد کی وضاحت کی ضرورت کو ختم کر دیتا ہے۔

-7 ایک طالب علم کو آن لائن لائبریری سسٹم ڈیزائن کرنے کا منصوبہ دیا گیا ہے۔ کمپیوٹیشنل تھنکنگ کے اصولوں کے مطابق، ان کا پہلا قدم کیا ہونا چاہیے؟

(A) وقت بچانے کے لیے فوراً کوڈنگ شروع کر دینا۔

(B) بنیادی مسئلے، صارف کی ضروریات، اور تکنیکی حدود کی شناخت کرنا۔

(C) ضروریات طے کرنے سے پہلے پروگرامنگ لینگویج کا انتخاب کرنا۔

(D) پورا سسٹم نافذ کرنے کے بعد فلو چارٹ بنانا۔

stepacademyofficial.com
https://stepacademyofficial.com/

8- ایک یم ایک ای کامرس ویب سائٹ تیار کر رہی ہے لیکن صارف کی غیر واضح ضروریات کی وجہ سے مشکلات کا سامنا ہے۔ اس مسئلے کا بہترین حل کیا ہوگا؟

(A) ترقیاتی عمل شروع کرنے سے پہلے صارف کی ضروریات کو واضح طور پر متعین کرنا۔

(B) صارف کی ضروریات کو نظر انداز کر کے صرف تکنیکی پہلوؤں پر توجہ دینا۔

(C) بغیر کسی منظم منصوبہ بندی کے آزمائش اور غلطی کے طریقے کو اپنانا۔

(D) پہلے ویب سائٹ تیار کر کے بعد میں صارف کی رائے کے مطابق تبدیلیاں کرنا۔

9- ایک تنظیم کو اپنے کام کے طریقہ کار میں غیر مؤثر حکمت عملی کا سامنا ہے۔ درج ذیل میں سے کون سا اصول اس صورتحال کو بہتر بنانے میں مدد دے سکتا ہے؟

(A) مسئلے کو آسان بنانے کے لیے اسے چھوٹے اور قابلِ حل حصوں میں تقسیم کرنا۔

(B) اضافی کام سے بچنے کے لیے دستاویزات کو ترک کر دینا۔

(C) لچک کو فروغ دینے کے لیے مقصد سازی کو ختم کر دینا۔ (D) بغیر کسی منصوبہ بندی کے بے ترتیب حل نافذ کرنا۔

10- ایک پروگرامر کو ایک ڈیٹا سیٹ میں سب سے بڑی تعداد تلاش کرنی ہے جس میں لاکھوں قدریں موجود ہیں۔ کمپیوٹیشنل تھنکنگ کے اصولوں کے مطابق، انہیں سب سے پہلے کس چیز پر غور کرنا چاہیے؟

(A) اس الگورتھم کو نافذ کرنے کے لیے استعمال ہونے والی پروگرامنگ لینگویج۔

(B) مختلف الگورتھمز کی وقت پیچیدگی (ٹائم کمپلیکسٹی) اور سب سے مؤثر الگورتھم کا انتخاب۔

(C) وقت بچانے کے لیے بغیر کسی منصوبہ بندی کے کوڈ لکھنا۔

(D) آزمائش اور غلطی کے طریقے سے سب سے بڑی تعداد تلاش کرنا۔

جوابات

1- (C)	2- (B)	3- (B)	4- (B)	5- (A)	6- (B)	7- (B)	8- (A)	9- (A)	10- (B)

محکمہ تعلیم کی نئی امتحانی تکنیکس (Knowledge, Understanding, Application, Analytical & Conceptual) کی روشنی میں مرتب کیے گئے مختصر جوابی سوالات

7.1 کمپیوٹیشنل تھنکنگ کی تعریف

☆ مختصر جواب دیں۔

1- کمپیوٹیشنل تھنکنگ کن شعبوں میں استعمال کی جاسکتی ہے؟

جواب: کمپیوٹیشنل تھنکنگ کمپیوٹر سائنس کے علاوہ حیاتیات، ریاضی، اور روزمرہ زندگی کے مختلف شعبوں میں استعمال کی جاسکتی ہے۔

2- کمپیوٹیشنل تھنکنگ روزمرہ زندگی میں کیسے مدد دیتی ہے؟

جواب: کمپیوٹیشنل تھنکنگ روزمرہ زندگی میں مسائل کو حل کرنے میں مدد دیتی ہے، جیسے سفر کی منصوبہ بندی یا روزانہ کے کاموں کو مؤثر طریقے سے منظم کرنا۔

3- فلو چارٹ کیا ہوتا ہے؟

جواب: فلو چارٹ ایک تصویری خاکہ ہوتا ہے جو کسی عمل یا سسٹم کے مختلف مراحل کو علامتوں اور تیروں کے ذریعے ظاہر کرتا ہے۔

stepacademyofficial.com

4- فلوچارٹس کن شعبوں میں زیادہ استعمال ہوتے ہیں؟

جواب: فلوچارٹس کمپیوٹر سائنس، انجینئرنگ، اور کاروبار میں زیادہ استعمال ہوتے ہیں۔

5- فلوچارٹس معلومات کی ترسیل میں کیسے مدد دیتے ہیں؟

جواب: فلوچارٹس پیچیدہ عمل کو آسانی سے سمجھنے کے قابل بناتے ہیں اور ایک وسیع ناظرین کے لیے واضح وضاحت فراہم کرتے ہیں۔

6- مسئلہ حل کرنے میں فلوچارٹس کیوں اہم ہیں؟

جواب: فلوچارٹس کسی عمل میں موجود رکاوٹوں اور کمزوریوں کو شناخت کرنے میں مدد دیتے ہیں، جس سے بہتر اور مؤثر حل نکالے جا سکتے ہیں۔

7- پیٹرن کی شناخت (Pattern Recognition) کیا ہے؟

جواب: پیٹرن کی شناخت کسی بھی مسئلے یا ڈیٹا میں مماثلتوں یا ترتیب کو پہچاننے کا عمل ہے، جو نتائج کو سمجھنے اور پیش گوئی کرنے میں مدد دیتا ہے۔

8- پیٹرن کی شناخت کی مدد سے مربع کا رقبہ کیسے معلوم کیا جا سکتا ہے؟

جواب: مربع کے رقبے کو معلوم کرنے کے لیے مسلسل طاق اعداد کو جوڑا جا سکتا ہے، جیسے کہ 3 کا مربع حاصل کرنے کے لیے: 1+3+5=9۔

9- تجدید (Abstraction) کیا ہے اور یہ کیوں اہم ہے؟

جواب: تجدید (Abstraction) وہ عمل ہے جس میں غیر ضروری تفصیلات کو نظر انداز کر کے صرف اہم معلومات پر توجہ دی جاتی ہے۔ یہ پیچیدہ مسائل کو آسان بنانے میں مدد دیتا ہے، جس سے ان کا حل نکالنا زیادہ مؤثر ہو جاتا ہے۔

10- تجدید (Abstraction) پیچیدہ مسائل کے حل میں کیسے مدد دیتی ہے؟

جواب: تجدید (Abstraction) کسی بڑے مسئلے کو چھوٹے اور آسان حصوں میں تقسیم کر کے صرف ضروری تفصیلات پر توجہ مرکوز رکھتی ہے، جس سے مسئلے کو زیادہ مؤثر طریقے سے حل کیا جا سکتا ہے۔

11- الگورتھم کو کیک بنانے کی ترکیب سے کیوں تشبیہ دی جاتی ہے؟

جواب: الگورتھم کو کیک بنانے کی ترکیب سے اس لیے تشبیہ دی جاتی ہے کیونکہ دونوں میں ایک خاص ترتیب میں دیے گئے اقدامات کو فالو کرنا ضروری ہوتا ہے تاکہ مطلوبہ نتیجہ حاصل کیا جا سکے۔

12- درخت لگانے کی مثال الگورتھم کے تصور کو کیسے واضح کرتی ہے؟

جواب: درخت لگانے کے عمل میں مخصوص اقدامات شامل ہوتے ہیں، جیسے جگہ کا انتخاب، گڑھا کھودنا، پودا لگانا، اور پانی دینا۔ یہ عمل ایک ترتیب شدہ ہدایت نامے کی طرح ہوتا ہے، جیسے کمپیوٹر میں الگورتھم۔

13- کمپیوٹر کے علاوہ روزمرہ زندگی میں الگورتھم کیسے استعمال ہوتے ہیں؟

جواب: الگورتھم روزمرہ زندگی میں ہر اس عمل میں استعمال ہوتے ہیں جو ایک مخصوص ترتیب کے ساتھ مکمل کیا جاتا ہے، جیسے دوست کے گھر کا راستہ تلاش کرنا، کسی گیم کے اصولوں پر عمل کرنا، یا دانت برش کرنے کا مرحلہ وار طریقہ اپنانا۔

7.2 کمپیوٹیشنل تھنکنگ کے اصول

14- مسئلے کو سمجھنا کمپیوٹیشنل تھنکنگ میں کیوں اہم ہے اور یہ مؤثر حل نکالنے میں کیسے مدد دیتا ہے؟

جواب: کسی بھی مسئلے کو حل کرنے کے لیے پہلے اس کا مکمل اور درست ادراک ضروری ہے۔ مسئلے کو صحیح طریقے سے سمجھنے سے غیر ضروری پیچیدگیاں کم ہوتی ہیں اور بہترین حل نکالنے میں مدد ملتی ہے۔

15- اگر ہم کسی مسئلے کو غلط سمجھیں تو اس کے کیا اثرات ہو سکتے ہیں؟

جواب: اگر مسئلے کو غلط سمجھا جائے تو حل بھی غلط ہو سکتا ہے، اور غیر ضروری محنت اور وقت ضائع ہو سکتا ہے۔ مثال کے طور پر، اگر کوئی طالب علم اسکول کی ویب سائٹ بنانے میں درکار فیچرز کو صحیح طریقے سے نہ سمجھے تو وہ غلط ڈیزائن بنا سکتا ہے۔

stepacademyofficial.com

https://stepacademyofficial.com/

16- مسئلے کے حل میں 'اہداف کی وضاحت' کا کیا کردار ہے؟

جواب: مسئلے کو حل کرنے کے لیے واضح اہداف کا تعین ضروری ہے۔ جب ہدف واضح ہوگا تو مسائل کو حل کرنے کی کوشش زیادہ منظم اور نتیجہ خیز ہوگی، جس سے وسائل کی بچت اور مؤثر حل حاصل کیے جا سکیں گے۔

17- کسی مسئلے کو آسان بنانے کے کیا فوائد ہیں؟

جواب: کسی پیچیدہ مسئلے کو چھوٹے حصوں میں تقسیم کرنے سے اسے بہتر طور پر سمجھا جا سکتا ہے اور اس کا حل آسانی سے نکالا جا سکتا ہے۔ اس سے توجہ مسائل کے اہم پہلوؤں پر مرکوز رہتی ہے اور نتیجہ زیادہ مؤثر ہوتا ہے۔

7.3	الگورتھم ڈیزائن کرنے کے طریقے
7.4	الگورتھم کی سرگرمیاں

18- الگورتھم ڈیزائن کے طریقوں کا مقصد کیا ہے؟

جواب: الگورتھم ڈیزائن کے طریقے مختلف مسائل کو مؤثر طریقے سے حل کرنے کے لیے تکنیکیں فراہم کرتے ہیں۔ یہ کسی بھی مسئلے کو سمجھنے اور اس کے لیے بہترین طریقہ منتخب کرنے میں مدد دیتے ہیں۔

19- فلو چارٹ مسئلہ حل کرنے کے لیے ایک مفید ٹول کیوں ہیں؟

جواب: فلو چارٹ ایک عمل یا نظام کے مراحل کو تصویری شکل میں ظاہر کرتے ہیں، جہاں مختلف علامات کو تیروں کے ذریعے جوڑا جاتا ہے۔ یہ کمپیوٹر سائنس، انجینئرنگ اور کاروبار سمیت کئی شعبوں میں استعمال ہوتے ہیں تاکہ پیچیدہ نظاموں کو آسان اور واضح طریقے سے پیش کیا جا سکے۔

20- فلو چارٹ کے استعمال کے اہم فوائد کیا ہیں؟

جواب: فلو چارٹ کسی عمل کو صاف اور مختصر طریقے سے پیش کرتے ہیں۔ یہ رابطے، مسئلہ حل کرنے اور دستاویزات کے لیے بہترین ٹول ہیں۔ اس کے علاوہ، یہ کسی بھی عمل میں موجود خامیوں اور غیر مؤثر حصوں کی نشاندہی کر کے بہتری میں مدد دیتے ہیں۔

7.6	لاجک آف الگورتھمز فار ریزولیوشن آف پرابلمز (LARP) کا تعارف
7.7	غلطی کی شناخت اور ڈیبگنگ

21- ایل اے آر پی (LARP) کیا ہے اور اسے الگورتھم سیکھنے کا انٹرایکٹو طریقہ کیوں کہا جاتا ہے؟

جواب: LARP کا مطلب ''Logic of Algorithms for Resolution of Problems'' ہے۔ یہ ایک ایسا طریقہ ہے جس میں الگورتھم کو عملی طور پر چلا کر ان کے نتائج دیکھے جاتے ہیں۔ یہ سیکھنے والوں کے لیے ایک تجرباتی میدان فراہم کرتا ہے جہاں وہ مختلف الگورتھمز کو آزما سکتے ہیں اور ڈیٹا پروسیسنگ کو بہتر طریقے سے سمجھ سکتے ہیں۔

22- ایل اے آر پی الگورتھم کے آؤٹ پٹ پر مختلف ان پٹس کے اثرات کو سمجھنے میں کیسے مدد دیتا ہے؟

جواب: ایل اے آر پی طلبا کو یہ دیکھنے کا موقع فراہم کرتا ہے کہ کس طرح مختلف ان پٹس الگورتھم کے آؤٹ پٹ کو متاثر کرتے ہیں۔ ان پٹس کو تبدیل کر کے طلبا نتائج میں تبدیلیوں کو دیکھ سکتے ہیں، جس سے انہیں ان پٹ اور الگورتھم کے رویے کے درمیان تعلق سمجھنے میں مدد ملتی ہے۔

23- ایل اے آر پی میں الگورتھم لکھنے کے لیے کون سے اہم اجزاء استعمال کیے جاتے ہیں؟

جواب: LARP میں الگورتھم لکھنے کے لیے درج ذیل اجزاء استعمال کیے جاتے ہیں:

1- START اور END: الگورتھم کے آغاز اور اختتام کی وضاحت کے لیے

2- WRITE: پیغامات کو ظاہر کرنے کے لیے

3- READ: صارف سے ان پٹ لینے کے لیے

stepacademyofficial.com

4- IF…THEN…ELSE: فیصلے کرنے کے عمل کے لیے

یہ اجزاء الگورتھم کو سمجھنے اور اس پر عمل درآمد کو آسان بناتے ہیں۔

24- ڈیبگنگ کیا ہے اور یہ الگورتھم اور فلو چارٹس میں کیوں ضروری ہے؟

جواب: ڈیبگنگ کا مطلب ہے کہ الگورتھم یا فلو چارٹ میں موجود غلطیوں کو تلاش کرنا اور درست کرنا۔ یہ ضروری ہے کیونکہ غلطیاں (bugs) الگورتھم کو درست طریقے سے کام کرنے سے روک سکتی ہیں، اور ڈیبگنگ ان غلطیوں کو ختم کر کے صحیح نتائج حاصل کرنے میں مدد دیتی ہے۔

25- سینٹیکس (Syntax) اور لاجیکل (Logical) غلطیوں میں کیا فرق ہے؟

جواب: **سینٹیکس غلطیاں:** جب الگورتھم میں کوئی چیز غلط لکھی جائے، جیسے کہ کوئی مرحلہ چھوڑ دینا یا غلط نشان (symbol) استعمال کرنا۔ یہ غلطیاں عام طور پر آسانی سے پکڑی جاتی ہیں کیونکہ LARP ٹول ان کی نشاندہی کر دیتا ہے۔

منطقی غلطیاں: جب الگورتھم کی منطق میں کوئی مسئلہ ہو، جس کی وجہ سے یہ غلط نتیجہ دے۔ یہ غلطیاں پکڑنا مشکل ہوتا ہے کیونکہ الگورتھم کام تو کرتا ہے لیکن صحیح جواب نہیں دیتا۔

26- رن ٹائم (Run-time) غلطیاں کیا ہوتی ہیں؟ مثال دیں۔

جواب: رن ٹائم غلطیاں اس وقت ہوتی ہیں جب الگورتھم چل رہا ہوتا ہے اور کوئی ناممکن عمل کرنے کی کوشش کرتا ہے۔ مثال کے طور پر، اگر کوئی عدد صفر سے تقسیم کیا جائے تو یہ ایک رن ٹائم غلطی ہوگی۔

27- منطقی غلطیاں سب سے مشکل کیوں ہوتی ہیں؟

جواب: لاجیکل غلطیاں مشکل ہوتی ہیں کیونکہ یہ سینٹیکس غلطیوں کی طرح فوراً ظاہر نہیں ہوتیں۔ الگورتھم چلتا رہتا ہے لیکن غلط نتائج دیتا ہے، جیسے کسی فیصلے میں غلط شرط (condition) استعمال کر لینا۔

28- وہ دو ڈیبگنگ تکنیک بتائیں جو الگورتھم میں غلطیاں تلاش کرنے میں مدد دیتی ہیں؟

جواب: 1- **مراحل کو ٹریس کرنا:** الگورتھم یا فلو چارٹ کے ہر مرحلے کو بغور دیکھنا تاکہ غلطی معلوم ہو سکے۔

2- **شرائط کو چیک کرنا:** تمام فیصلوں میں استعمال ہونے والی شرائط کو دوبارہ جانچنا تاکہ لاجیکل غلطیوں سے بچا جا سکے۔

29- LARP میں "Undefined Variable" غلطی کا کیا مطلب ہے؟

جواب: یہ غلطی اس وقت آتی ہے جب الگورتھم میں کوئی ایسا ویری ایبل (variable) استعمال کیا جاتا ہے جو پہلے بیان (define) نہیں کیا گیا ہوتا۔

30- الگورتھم میں تبصرے (comments) لکھنے سے ڈیبگنگ میں کیسے مدد ملتی ہے؟

جواب: الگورتھم میں تبصرے لکھنے سے ہر حصے کی وضاحت ہو جاتی ہے، جس سے غلطیوں کو پکڑنا اور منطق کو سمجھنا آسان ہو جاتا ہے۔

31- اگر کوئی الگورتھم بہت پیچیدہ ہو اور ڈیبگ کرنا مشکل ہو تو پروگرامر کو کیا کرنا چاہیے؟

جواب: اگر الگورتھم بہت پیچیدہ ہو تو پروگرامر کو مسئلے کو آسان بنانے کے لیے اسے چھوٹے حصوں میں تقسیم کر کے ہر حصے کو علیحدہ علیحدہ جانچنا چاہیے۔

مختصر جوابی کنسیپچوئل (Conceptual) سوالات

☆ **مختصر جواب دیں۔**

1- ڈی کمپوزیشن (Decomposition) پیچیدہ مسائل کو حل کرنے میں کیسے مدد دیتی ہے؟

جواب: ڈی کمپوزیشن ایک پیچیدہ مسئلے کو چھوٹے، آسان حصوں میں تقسیم کر دیتی ہے۔ اس سے ہر حصے کو الگ الگ حل کرنا ممکن ہو جاتا ہے، جس سے پورے مسئلے کو حل کرنا آسان ہو جاتا ہے۔

stepacademyofficial.com

https://stepacademyofficial.com/

-2 پرندوں کے لیے برڈ ہاؤس (Bird house) بناتے وقت مختلف پرندوں کے سوراخوں کے سائز پر تحقیق کیوں ضروری ہے؟

جواب: ہر پرندے کی نسل کے لیے مختلف سائز کے داخلے (Entrance) کی ضرورت ہوتی ہے۔ اگر صحیح سائز کا سوراخ بنایا جائے تو پرندے آسانی سے اندر جاسکتے ہیں، اور برڈ ہاؤس ان کے لیے موزوں ہوگا۔

-3 مسئلے کو آسان بنانے (Problem Simplification) سے مؤثر حل تلاش کرنے میں کیسے مدد ملتی ہے؟ ایک مثال دیں۔

جواب: مسئلے کو آسان بنانے کا مطلب یہ ہے کہ پیچیدہ مسئلے کو چھوٹے، قابلِ حل حصوں میں تقسیم کیا جائے۔ اس سے ہر حصہ الگ الگ سمجھنا اور حل کرنا آسان ہو جاتا ہے، جس سے پورا مسئلہ مؤثر طریقے سے حل ہو جاتا ہے۔

مثال کے طور پر، اگر ہم ایک ویب سائٹ ڈیزائن کر رہے ہیں تو ہم اسے مختلف حصوں میں تقسیم کر سکتے ہیں جیسے کہ لے آؤٹ ڈیزائن، مواد کی تخلیق، اور فنکشنل کوڈنگ۔

-4 فلو چارٹ (Flow chart) کو الگورتھم ڈیزائن میں ایک مؤثر ٹول کیوں سمجھا جاتا ہے؟ اس کی اہمیت بیان کریں۔

جواب: فلو چارٹ ایک الگورتھم کو علامتوں (Symbols) اور تیروں (Arrows) کے ذریعے بصری طور پر پیش کرتے ہیں، جس سے پیچیدہ کاموں کو آسانی سے سمجھا جاسکتا ہے۔ یہ کمپیوٹر سائنس، انجینئرنگ اور کاروبار میں استعمال کیے جاتے ہیں تاکہ مختلف مراحل کو ماڈل کیا جاسکے اور انہیں واضح طریقے سے پیش کیا جاسکے۔

فلو چارٹس کی اہمیت درج ذیل وجوہات کی بناء پر ہے:

- وضاحت (Clarity) فراہم کرنا
- مؤثر ابلاغ (Communication) ممکن بنانا
- مسائل کے حل (Problem-solving) میں مدد دینا
- ریکارڈ رکھنے (Documentation) کے لیے استعمال ہونا

-5 سائنسی تجربات میں سیمولیشن (Simulation) مختلف حالات کو جانچنے میں کیسے مدد دیتی ہے، اور بعض صورتوں میں یہ حقیقی تجربات سے بہتر کیوں ہوتی ہے؟

جواب: سیمولیشن ہمیں مختلف حالات کو حقیقی تجربہ کیے بغیر جانچنے کی سہولت فراہم کرتی ہے۔ مثال کے طور پر، اگر ہم پودوں کی نشوونما پر تحقیق کر رہے ہیں، تو سیمولیشن کے ذریعے مختلف مقدار میں پانی اور دھوپ کے اثرات کو بغیر حقیقی تجربے کے معلوم کیا جاسکتا ہے۔

یہ حقیقی تجربات کے مقابلے میں زیادہ مؤثر ہوتی ہے کیونکہ:

- یہ کم لاگت (Cost-effective) ہوتی ہے
- یہ محفوظ (Safe) ہوتی ہے
- مختلف شرائط کو بار بار (Repeatable) آزمایا جاسکتا ہے

-6 فلو چارٹ، سوڈوکوڈ (Pseudocode) کے مقابلے میں الگورتھم کے بہاؤ کو زیادہ مؤثر طریقے سے کیسے سمجھاتا ہے، اور اس میں استعمال ہونے والی علامتیں اس میں کیسے مدد دیتی ہیں؟

جواب: فلو چارٹ میں مستطیل، ہیرے اور بیضوی شکلوں کے ساتھ تیروں کا استعمال کیا جاتا ہے، جو الگورتھم کے بہاؤ کو بصری طور پر پیش کرتے ہیں۔ اس کا گرافیکل انداز دیکھنے والے کو الگورتھم کی ساخت اور فیصلوں کو فوری طور پر سمجھنے میں مدد دیتا ہے۔

سوڈوکوڈ صرف متن پر مبنی ہوتا ہے، جبکہ فلو چارٹ ایک فلم کی طرح الگ الگ مراحل کو ظاہر کرتا ہے، جس سے اس کا سمجھنا زیادہ آسان ہو جاتا ہے۔

stepacademyofficial.com

-7 ایل اے آر پی (LARP) الگورتھم سیکھنے کے عمل کو کس طرح آسان بناتا ہے اور یہ سیکھنے والوں کو مسئلہ حل کرنے اور منطقی سوچنے میں کس طرح مدد دیتا ہے؟

جواب: ایل اے آر پی ایک واضح اور منظم طریقہ فراہم کرتا ہے، جس میں سیکھنے والے الگورتھم کو چلا کر اس کے نتائج دیکھ سکتے ہیں۔ اس میں آسان سینٹیکس (Syntax) استعمال ہوتی ہے، جیسے:

-1 START اور END الگورتھم کے آغاز اور اختتام کے لیے

-2 WRITE پیغام ظاہر کرنے کے لیے

-3 READ ان پٹ لینے کے لیے

-4 IF...THEN...ELSE فیصلے کرنے کے لیے

یہ طریقہ مسئلے کو چھوٹے، قابلِ حل حصوں میں تقسیم کرتا ہے اور طلباء کو منطق پر زیادہ توجہ دینے کی ترغیب دیتا ہے، جس سے ان کی مسئلہ حل کرنے اور منطقی سوچنے کی صلاحیت بہتر ہوتی ہے۔

-8 ایل اے آر پی میں مختلف ان پٹس کے ساتھ تجربہ کرنے کی صلاحیت کیوں ضروری ہے اور یہ الگورتھم کے رویے کو سمجھنے میں کیسے مدد دیتی ہے؟

جواب: ایل اے آر پی میں مختلف ان پٹس آزمانا ضروری ہے کیونکہ اس سے سیکھنے والے یہ دیکھ سکتے ہیں کہ الگورتھم کے آؤٹ پٹ پر ان پٹس کیسے اثر انداز ہوتے ہیں۔ اگر ان پٹس کو بدلا جائے اور نتائج کو دیکھا جائے، تو اس سے غلطیوں کی نشاندہی کی جا سکتی ہے، منطق کو بہتر بنایا جا سکتا ہے، اور مسئلہ حل کرنے کی مہارت میں بہتری آ سکتی ہے۔ اس عمل سے نہ صرف سمجھ بوجھ میں اضافہ ہوتا ہے بلکہ الگورتھم ڈیزائن اور ڈیبگنگ میں بھی مدد ملتی ہے۔

-9 ڈیبگنگ (Debugging) الگورتھم ڈویلپمنٹ کا ایک لازمی حصہ کیوں ہے؟ وضاحت کریں کہ 'Trace the steps' اور 'Simplify the Problem' تکنیکیں غلطیوں کی نشاندہی اور ان کو درست کرنے میں کیسے مدد دیتی ہیں؟

جواب: ڈیبگنگ کا مطلب ہے کہ الگورتھم میں موجود غلطیوں کو تلاش کرنا اور درست کرنا۔

'Trace the steps' میں الگورتھم کے ہر مرحلے کو ترتیب سے دیکھا جاتا ہے تاکہ یہ معلوم ہو سکے کہ غلطی کہاں ہو رہی ہے۔

'Simplify the Problems' میں مسئلے کو چھوٹے حصوں میں تقسیم کر کے ہر حصہ الگ سے جانچا جاتا ہے، جس سے غلطیوں کو تلاش کرنا اور درست کرنا آسان ہو جاتا ہے۔

یہ تکنیکیں الگورتھم کی درستگی کو یقینی بنانے میں مدد دیتی ہیں۔

-10 منطقی (Logical) غلطیاں تلاش کرنا مشکل کیوں ہوتا ہے اور یہ سینٹیکس (Syntax) غلطیوں سے کیسے مختلف ہوتی ہیں؟

جواب: سینٹیکس غلطیاں تب ہوتی ہیں جب الگورتھم کے لکھنے کے اصولوں کی خلاف ورزی کی جاتی ہے، جیسے غلط کمانڈز یا غائب علامتیں۔ یہ غلطیاں آسانی سے پکڑی جا سکتی ہیں کیونکہ ایل اے آر پی ٹول انہیں خود ہی نشان زد کر دیتا ہے۔

لاجیکل غلطیاں تب ہوتی ہیں جب الگورتھم کی منطق میں کوئی خرابی ہو، جس کی وجہ سے یہ غلط نتائج دیتا ہے، حالانکہ وہ بغیر کسی خرابی کے چلتا رہتا ہے۔ چونکہ یہ غلطیاں سسٹم میں کسی خرابی کا پیغام نہیں دیتیں، اس لیے انہیں تلاش کرنا زیادہ مشکل ہوتا ہے۔

خلاصہ

(i) کمپیوٹیشنل تھنکنگ ایک اہم مہارت ہے جو افراد کو ایسے طریقوں کا استعمال کرتے ہوئے پیچیدہ مسائل کو حل کرنے کے قابل بناتی ہے جو کمپیوٹر سائنس میں شامل عمل کی عکاسی کرتے ہیں۔

(ii) ڈی کمپوزیشن ایک پیچیدہ مسئلے کو چھوٹے، زیادہ منظم حصوں میں توڑنے کا عمل ہے۔

(iii) پیٹرن کی شناخت میں مسائل کے درمیان اور ان کے اندر مماثلت یا نمونے تلاش کرنا شامل ہے۔

stepacademyofficial.com

https://stepacademyofficial.com/

(iv) تجدید میں پیچیدہ مسائل کو چھوٹے اور زیادہ منظم حصوں میں تقسیم کر کے آسان بنانا اور غیر ضروری مسائل کو نظر انداز کرتے ہوئے صرف ضروری تفصیلات پر توجہ مرکوز کرنا شامل ہے۔

(v) الگورتھم کسی مسئلے کو حل کرنے یا کسی کام کو مکمل کرنے کے لیے ہدایات کا مرحلہ وار مجموعہ ہے۔

(vi) مسئلے کو سمجھنا مسئلے کو حل کرنے میں پہلا اور سب سے اہم مرحلہ ہے، خاص طور پر کمپیوٹیشنل تھنکنگ میں۔

(vii) کسی مسئلے کو آسان بنانے میں اسے چھوٹے، زیادہ منظم ذیلی مسائل میں توڑنا شامل ہے۔

(viii) بہترین حل کا انتخاب کرنے میں مختلف طریقوں کا جائزہ لینا اور سب سے زیادہ موثر کا انتخاب کرنا شامل ہے۔

(ix) فلو چارٹ کسی عمل یا نظام میں مراحل کی بصری نمائندگی ہیں، جو تیروں سے منسلک مختلف علامتوں کا استعمال کرتے ہوئے دکھایا گیا ہے۔

(x) سوڈوکوڈ سادہ اور غیر رسمی زبان کا استعمال کرتے ہوئے الگورتھم کی نمائندگی کرنے کا ایک طریقہ ہے جسے سمجھنا آسان ہے۔ یہ پروگرامنگ زبانوں کی ساخت کو سادہ انگریزی زبان پڑھنے کی قابلیت کے ساتھ جوڑتا ہے، جس سے یہ الگورتھم کی منصوبہ بندی اور وضاحت کے لیے ایک مفید آلہ بن جاتا ہے۔

(xi) ٹائم کمپلیکسٹی اس بات کی پیمائش کرنے کا ایک طریقہ ہے کہ الگورتھم کتنی تیز یا سست کارکردگی کا مظاہرہ کرتا ہے۔ یہ ہمیں بتاتا ہے کہ ان پٹ کے سائز میں اضافے کے ساتھ الگورتھم کا چلنے کا وقت کیسے تبدیل ہوتا ہے۔

(xii) سپیس کمپلیکسٹی ان پٹ سائز کے سلسلے میں الگورتھم استعمال ہونے والی میموری کی مقدار کی پیمائش کرتی ہے۔ ان پٹ کے لیے ضروری میموری اور الگورتھم کے ذریعے استعمال ہونے والی کسی بھی اضافی میموری دونوں پر غور کرنا ضروری ہے۔

(xiii) ڈرائی رن میں کسی بھی غلطی کی نشاندہی کرنے کے لیے نمونے کے اعداد و شمار کے ساتھ الگورتھم کے ذریعے دستی طور پر جانا شامل ہے۔

(xiv) سیمولیشن اس وقت ہوتا ہے جب ہم حقیقی دنیا کے عمل یا نظام کا ماڈل بنانے کے لیے کمپیوٹر پروگراموں کا استعمال کرتے ہیں۔

(xv) LARP کا مطلب ہے لائیو الگورتھم چلانا اور مشق کرنا۔ یہ سیکھنے کا ایک دلچسپ اور انٹرایکٹو طریقہ ہے جو سیکھنے والوں کو الگورتھم کو عملی طور پر چلانے اور ان کے نتائج دیکھنے میں مدد دیتا ہے۔

(xvi) ڈیبگنگ الگورتھم یا فلو چارٹ میں غلطیوں کو تلاش کرنے اور ٹھیک کرنے کا عمل ہے۔

حل مشقی سوالات

1۔ درست جواب کا انتخاب کریں:

(i) **مندرجہ ذیل میں سے کمپیوٹیشنل تھنکنگ کی بہترین تعریف کون سی ہے؟**

(الف) صرف ریاضیاتی حساب کتاب کا استعمال کرتے ہوئے مسائل کو حل کرنے کا ایک طریقہ ہے۔

(ب) ایک مسئلہ حل کرنے کا طریقہ جو منظم، الگورتھمز اور منطقی تھنکنگ کو استعمال کرتا ہے۔

(ج) ایک تکنیک جو خاص طور پر کمپیوٹر پروگرامنگ میں استعمال ہوتی ہے۔

(د) ایک ایسا طریقہ جو حقیقی دنیا کی ایپلی کیشنز کو نظر انداز کرتا ہے۔

(ii) **کمپیوٹیشنل تھنکنگ میں مسئلہ کی تقسیم کیوں اہم ہے؟**

(الف) یہ مسائل کو چھوٹے، زیادہ منظم حصوں میں تقسیم کر کے آسان بناتا ہے۔

(ب) یہ مزید تفصیلات شامل کر کے مسائل کو پیچیدہ بناتا ہے۔ (ج) یہ مسئلے کو حل کرنے کی ضرورت کو ختم کرتا ہے۔

(د) یہ صرف سادہ مسائل کے لیے مفید ہے۔

stepacademyofficial.com

(iii) نمونے کی شناخت میں کیا شامل ہیں؟

(الف) مسائل کے اندر مماثلت تلاش کرنا اور ان کا فائدہ اُٹھانا (ب) تکرار کے عناصر کو نظر انداز کرنا

(ج) مسائل کو چھوٹے چھوٹے حصوں میں تقسیم کرنا (د) تفصیلی الگورتھم لکھنا

(iv) کون سی اصطلاح بنیادی خیال پر توجہ مرکوز کرنے کے لیے تفصیلات کو نظر انداز کرنے کے عمل پر مشتمل ہے؟

(الف) ڈی کمپوزیشن (ب) پیٹرن کی شناخت (ج) تجدید (د) الگورتھم ڈیزائن

(v) مندرجہ ذیل میں سے کون سا کمپیوٹیشنل تھنکنگ کا بنیادی اصول ہے؟

(الف) مسئلے کی تفہیم کو نظر انداز کرنا (ب) مسئلے کو آسان بنانا

(ج) حل کے ڈیزائن سے گریز کرنا (د) بے ترتیب حل کو نافذ کرنا

(vi) الگورتھم کیا ہوتے ہیں؟

(الف) ڈیٹا کی فہرستیں (ب) گرافیکل نمائندگی

(ج) کسی مسئلے کو حل کرنے کے لیے مرحلہ وار ہدایات (د) بار بار دہرائے جانے والے پیٹرن

(vii) کمپیوٹیشنل تھنکنگ کے مطابق مسئلہ حل کرنے میں مندرجہ ذیل میں سے کون سا پہلا قدم ہے؟

(الف) حل لکھنا (ب) مسئلے کو سمجھنا (ج) فلوچارٹ ڈیزائن کرنا (د) حل کا انتخاب کرنا

(viii) فلوچارٹ کس لیے استعمال کیے جاتے ہیں؟

(الف) پروگرام کوڈ کرنے کے لیے (ب) گرافک طور پر الگورتھم کو ظاہر کرنے کے لیے

(ج) ریاضیاتی مساوات کو حل کرنے کے لیے (د) پیٹرنز کی شناخت کے لیے

(ix) سوڈو کوڈ کیا ہے؟

(الف) فلوچارٹ کی ایک قسم

(ب) سادہ زبان کا استعمال کرتے ہوئے الگورتھم کی ایک اعلیٰ سطحی وضاحت

(ج) ایک پروگرامنگ زبان (د) ایک ڈیبگنگ ٹول

(x) فلوچارٹ کو ڈرائی رن کرنے میں کیا شامل ہوتا ہے؟

(الف) پروگرامنگ زبان میں کوڈ لکھنا (ب) نمونہ ڈیٹا کے ساتھ فلوچارٹ کی جانچ

(ج) فلوچارٹ کو سوڈو کوڈ میں تبدیل کرنا (د) فلوچارٹ کی تفصیلات کو نظر انداز کرنا

جوابات

(i) (ب)	(ii) (الف)	(iii) (الف)	(iv) (ج)	(v) (ب)	(vi) (د)	(vii) (ب)	(viii) (ب)	(ix) (ب)	(x) (ج)

-2 مختصر جواب دیں۔

(i) کمپیوٹیشنل تھنکنگ کی وضاحت کریں۔

جواب: کمپیوٹیشنل تھنکنگ ایک مسئلہ حل کرنے کا طریقہ ہے جس میں پیچیدہ مسائل کو چھوٹے حصوں میں تقسیم کیا جاتا ہے، پیٹرن (patterns) پہچانے جاتے ہیں، غیر ضروری تفصیلات ہٹائی جاتی ہیں اور ایسے مرحلہ وار حل بنائے جاتے ہیں جو کمپیوٹر پر عملدرآمد کر سکے۔

(ii) کمپیوٹیشنل تھنکنگ میں Decomposition کیا ہے؟

جواب: ڈی کمپوزیشن وہ عمل ہے جس میں ایک پیچیدہ مسئلے کو چھوٹے اور قابلِ فہم حصوں میں تقسیم کیا جاتا ہے تاکہ اسے آسانی سے سمجھا اور حل کیا جا سکے۔

stepacademyofficial.com

https://stepacademyofficial.com/

(iii) Pattern Recognition کو ایک مثال کے ساتھ بیان کریں۔

جواب: پیٹرن ریکگنیشن کا مطلب ہے کہ مسائل میں یکساں خصوصیات یا رجحانات (trends) پہچاننا تا کہ مؤثر حل نکالا جا سکے۔

مثال: ریاضی میں، ضرب کے جدول کا پیٹرن پہچاننے سے سوالات جلدی اور آسانی سے حل کیے جا سکتے ہیں۔

(iv) مسئلہ حل کرنے میں تجرید اور اس کی اہمیت کی وضاحت کریں۔

جواب: ایبسٹریکشن غیر ضروری تفصیلات کو نظر انداز کر کے صرف اہم معلومات پر توجہ دینے کا عمل ہے۔ یہ پیچیدہ مسائل کو آسان بنانے اور مؤثر حل نکالنے میں مدد دیتا ہے۔

(v) الگورتھم کیا ہے؟

جواب: الگورتھم کسی خاص کام یا مسئلے کو حل کرنے کے لیے ایک مرحلہ وار ہدایات کا مجموعہ ہوتا ہے۔

(vi) مسئلہ کی تفہیم کمپیوٹیشنل تھنکنگ میں کس طرح مدد کرتی ہے؟

جواب: مسئلے کو سمجھنے سے اس کے بنیادی اجزاء کی شناخت، صحیح تکنیکوں کا انتخاب اور ایک مؤثر حل تیار کرنا آسان ہو جاتا ہے۔

(vii) فلو چارٹ کیا ہیں اور وہ کس طرح استعمال ہوتے ہیں؟

جواب: فلو چارٹ ایک ایسا خاکہ ہوتا ہے جو الگورتھم کو مختلف علامات اور تیروں کی مدد سے ظاہر کرتا ہے۔ یہ کسی عمل کے مراحل کو بصری طور پر واضح کرنے کے لیے استعمال کیا جاتا ہے تاکہ اسے بہتر طریقے سے سمجھا جا سکے۔

(viii) سوڈوکوڈ کے مقصد کی وضاحت کریں۔

جواب: سوڈوکوڈ ایک آسان اور ترتیب شدہ طریقہ ہے جس میں الگورتھم کو لکھا جاتا ہے۔ اس میں پروگرامنگ زبان سے ملتے جلتے الفاظ استعمال کیے جاتے ہیں تاکہ اسے سمجھنا اور عملی طور پر نافذ کرنا آسان ہو۔

(ix) آپ فلو چارٹ اور سوڈوکوڈ کے درمیان فرق کیسے کرتے ہیں؟

جواب: فلو چارٹ میں مختلف علامات اور خاکوں کے ذریعے کسی عمل کو ظاہر کیا جاتا ہے۔

سوڈوکوڈ میں تحریری اور ساختی جملوں کے ذریعے الگورتھم بیان کیا جاتا ہے۔

(x) ڈرائی رن کیا ہے اور یہ کیوں ضروری ہے؟

جواب: ڈرائی رن ایک ایسا عمل ہے جس میں کسی الگورتھم کو کمپیوٹر پر چلانے سے پہلے دستی طور پر مرحلہ وار جانچا جاتا ہے۔ اس کا مقصد غلطیوں کا پتا لگانا اور ان کو درست کرنا ہوتا ہے۔

(xi) LARP کیا ہے اور الگورتھم سیکھنے میں اس کی اہمیت کی وضاحت کریں۔

جواب: LARP (Learn Alogorithm Representation Process) ایک ایسا طریقہ ہے جو الگورتھم کو بصری شکل میں پیش کر کے سیکھنے میں مدد دیتا ہے۔ یہ منطقی بہاؤ (Logical Flow) کو سمجھنے اور غلطیوں کو درست کرنے میں معاون ثابت ہوتا ہے۔

(xii) دو ڈیبگنگ تکنیکس لکھیں اور ان کی وضاحت کریں۔

جواب: 1- اسٹیپس کا جائزہ لیں: الگورتھم کے ہر قدم کو غور سے چیک کریں تاکہ غلطی کا پتا چل سکے۔

2- شرائط چیک کریں: فیصلہ سازی کے مراحل میں تمام شرائط کو یقینی بنائیں کہ وہ درست ہیں۔

3- تفصیلی سوالات:

(i) ایک الگورتھم لکھیں جو کسی طالب علم کے حاصل کردہ نمبروں کی بنیاد پر گریڈ تفویض کرے۔ درج ذیل گریڈنگ سسٹم کا استعمال کریں:

- 90 اور اس سے زیادہ : A+
- 80 سے 89 : A

stepacademyofficial.com

- 70 سے 79 : B
- 60 سے 69 : C
- 60 سے نیچے : F

جواب: جواب کے لیے دیکھیے سوال نمبر 8

(ii) وضاحت کریں کہ آپ ایک پیچیدہ کمپیوٹیشنل مسئلے کو حل کرنے کے لیے الگورتھم ڈیزائن کے طریقوں، جیسے فلو چارٹ اور سوڈو کوڈ کا استعمال کیسے کریں گے۔ ایک تفصیلی مثال کے ساتھ اپنی وضاحت کو بیان کریں۔

جواب: جواب کے لیے دیکھیے سوال نمبر 9

(iii) کمپیوٹیشنل تھنکنگ کی وضاحت کریں اور جدید مسائل کو حل کرنے میں اس کی اہمیت کو بیان کریں۔ مثالیں فراہم کریں کہ مختلف شعبوں میں کمپیوٹیشنل تھنکنگ کو کس طرح لاگو کیا جا سکتا ہے۔

جواب: جواب کے لیے دیکھیے سوال نمبر 1

(iv) کمپیوٹیشنل تھنکنگ میں تقسیم کے تصور پر تبادلہ خیال کریں۔ یہ کیوں اہم ہے؟

جواب: جواب کے لیے دیکھیے سوال نمبر 1

(v) کمپیوٹیشنل تھنکنگ کے حوالے سے پیٹرن کی شناخت کی وضاحت کریں۔ یہ مسئلہ حل کرنے میں کس طرح مدد کرتی ہے؟

جواب: جواب کے لیے دیکھیے سوال نمبر 2

(vi) کمپیوٹیشنل تھنکنگ میں Abstraction کیا ہے؟ اس کی اہمیت پر تبادلہ خیال کریں اور اس کی مثالیں فراہم کریں کہ پیچیدہ مسائل کو آسان بنانے کے لیے کس طرح اس کا استعمال کیا جا سکتا ہے۔

جواب: جواب کے لیے دیکھیے سوال نمبر 3

(vii) الگورتھم کیا ہے اس کی وضاحت کریں اور کمپیوٹیشنل تھنکنگ میں اس کے کردار کی وضاحت کریں۔ کسی مخصوص مسئلے کو حل کرنے کے لیے الگورتھم کی تفصیلی مثال فراہم کریں اور متعلقہ فلو چارٹ بنائیں۔

جواب: جواب کے لیے دیکھیے سوال نمبر 4

(viii) الگورتھم ڈیزائن کے طریقوں کے طور پر فلو چارٹ اور سوڈو کوڈ کا موازنہ کریں۔ ہر طریقہ کار کے فوائد اور نقصانات پر تبادلہ خیال کریں، اور ایسی مثالیں فراہم کریں جہاں ایک کو دوسرے پر ترجیح دی جا سکتی ہے۔

جواب: جواب کے لیے دیکھیے سوال نمبر 10

(ix) فلو چارٹ اور سوڈو کوڈ دونوں کے حوالے سے ڈرائی رن کے تصور کی وضاحت کریں۔ ڈرائی رن کرنے سے الگورتھم کے درست ہونے کی تصدیق کرنے میں کس طرح مدد ملتی ہے؟

جواب: جواب کے لیے دیکھیے سوال نمبر 12

(x) LARP کیا ہے؟ الگورتھم سیکھنے اور مشق کرنے میں اس کی اہمیت پر تبادلۂ خیال کریں۔

جواب: جواب کے لیے دیکھیے سوال نمبر 14

(xi) LARP کمپیوٹیشنل تھنکنگ کے اصولوں کی تفہیم اور اطلاق کو کس طرح بہتر بناتا ہے؟ ایسی مثال دیں کہ جہاں LARP کو الگورتھم کو بہتر بنانے کے لیے استعمال کیا جا سکتا ہے۔

جواب: جواب کے لیے دیکھیے سوال نمبر 15.

stepacademyofficial.com

https://stepacademyofficial.com/