کمپیوٹر سائنس نہم

یونٹ 7: کمپیوٹیشنل تھنکنگ

سوال نمبر 1: کمپیوٹیشنل تھنکنگ کی وضاحت کریں۔

جواب: کمپیوٹیشنل تھنکنگ (Computational Thinking) ایک مسئلہ حل کرنے کا عمل ہے جس میں ایسے طریقے اور تکنیک شامل ہیں جو کمپیوٹر سائنس کے اصولوں سے مطابقت رکھتے ہیں۔ یہ صرف کمپیوٹر سائنسدانوں کے لیے نہیں، بلکہ ہر اس شخص کے لیے ایک ضروری مہارت ہے جو پیچیدہ مسائل کو منظم اور منطقی طریقے سے حل کرنا چاہتا ہے۔ اس میں مسائل کو چھوٹے حصوں میں تقسیم کرنا، پیٹرن تلاش کرنا، غیر ضروری تفصیلات کو نظر انداز کرنا، اور مرحلہ وار حل تیار کرنا شامل ہے۔

سوال نمبر 2: کمپیوٹیشنل تھنکنگ کے اہم اجزاء کے نام لکھیں۔

جواب: کمپیوٹیشنل تھنکنگ کے چار اہم اجزاء ہیں:

1. **تجزیہ / تقسیم :(Decomposition)** ایک پیچیدہ مسئلے کو چھوٹے، قابل انتظام حصوں میں توڑنا۔
2. **پیٹرن کی شناخت :(Pattern Recognition)** مسائل کے اندر یا ان کے درمیان مماثلت یا پیٹرن تلاش کرنا۔
3. **تجرید :(Abstraction)** غیر ضروری تفصیلات کو نظر انداز کرتے ہوئے صرف اہم معلومات پر توجہ مرکوز کرنا۔
4. **الگورتھم :(Algorithms)** مسئلے کو حل کرنے کے لیے مرحلہ وار ہدایات کا ایک مجموعہ تیار کرنا۔

سوال نمبر3: کمپیوٹیشنل تھنکنگ میں decomposition (تجزیہ) کیا ہے؟

جواب: تجزیہ (Decomposition) کمپیوٹیشنل تھنکنگ کا وہ جزو ہے جس میں ایک بڑے اور پیچیدہ مسئلے کو چھوٹے، زیادہ قابل انتظام اور آسان کاموں میں تقسیم کیا جاتا ہے۔ اس سے ہر چھوٹے حصے پر الگ الگ توجہ مرکوز کرنا اور اسے حل کرنا آسان ہو جاتا ہے، جس سے مجموعی طور پر بڑا مسئلہ حل ہو جاتا ہے۔ مثال کے طور پر، ایک پرندے کا گھر بنانے کے کام کو ڈیزائننگ، مواد جمع کرنے، لکڑی کاٹنے، اور جوڑنے جیسے چھوٹے کاموں میں تقسیم کیا جا سکتا ہے۔

سوال نمبر4: تقسیم یا تجزیہ کی اہمیت بیان کریں۔

جواب: تقسیم یا تجزیہ (Decomposition) کی اہمیت یہ ہے کہ یہ پیچیدہ اور مشکل مسائل کو آسان بنا دیتا ہے۔ جب ایک بڑا مسئلہ چھوٹے حصوں میں تقسیم ہو جاتا ہے، تو اسے سمجھنا اور اس کا حل تلاش کرنا زیادہ آسان ہو جاتا ہے۔ یہ طریقہ کار ہمیں ہر حصے پر انفرادی طور پر توجہ مرکوز کرنے کی اجازت دیتا ہے، جس سے غلطیوں کا امکان کم ہوتا ہے اور مسئلہ حل کرنے کا عمل زیادہ منظم اور موثر ہو جاتا ہے۔

سوال نمبر5 Pattern recognition: (پیٹرن کی شناخت) کو ایک مثال کے ساتھ بیان کریں۔

جواب: پیٹرن کی شناخت (Pattern Recognition) وہ عمل ہے جس میں ڈیٹا یا مسائل کے اندر مماثلت، تکرار یا رجحانات کو تلاش کیا جاتا ہے۔ ان پیٹرنز کو پہچاننے سے ہمیں مسئلے کو زیادہ مؤثر طریقے سے حل کرنے میں مدد ملتی ہے۔ **مثال:** مربعوں کے رقبے میں ایک پیٹرن ہوتا ہے۔ اگر ایک مربع کی سائیڈ کی لمبائی 4,3,2,1 ہے، تو اس کا رقبہ 16,9,4,1 ہو گا۔ یہاں پیٹرن یہ ہے کہ ہر مربع کا رقبہ اس کی سائیڈ کی لمبائی کا مربع (side $length^2$) ہے، یا اسے مسلسل طاق اعداد کو جمع کر کے بھی حاصل کیا جا سکتا ہے) جیسے 3x3 مربع کا رقبہ 1+3+5=9 ہے(۔

سوال نمبر6: مسئلہ حل کرنے میں تجرید (abstraction) اور اس کی اہمیت کی وضاحت کریں۔

جواب: تجرید :(Abstraction) یہ وہ عمل ہے جس میں پیچیدہ نظاموں کو آسان بنانے کے لیے غیر ضروری تفصیلات کو نظر انداز کرتے ہوئے صرف ضروری اور متعلقہ معلومات پر توجہ مرکوز کی جاتی ہے۔ **اہمیت :** تجرید کی اہمیت یہ ہے کہ یہ ہمیں پیچیدگی کو کم کرنے میں مدد دیتی ہے۔ اس سے ہم مسئلے کی بنیادی ساخت کو سمجھ سکتے ہیں اور غیر ضروری تفصیلات میں الجھے بغیر ایک اعلیٰ سطحی حل تیار کر سکتے ہیں۔ مثال کے طور پر، چائے بنانے کے عمل میں ہم "پانی ابالیں" اور "چائے کی پتی ڈالیں" جیسے اہم مراحل پر توجہ دیتے ہیں اور کیتلی کے کام کرنے کے اندرونی میکانزم جیسی تفصیلات کو نظر انداز کر دیتے ہیں۔

سوال نمبر7: الگورتھم کیا ہے؟

جواب: الگورتھم کسی مسئلے کو حل کرنے یا کسی کام کو مکمل کرنے کے لیے مرحلہ وار ہدایات کا ایک واضح اور منظم مجموعہ ہے۔ یہ ایک ترکیب کی طرح ہوتا ہے جو بتاتا ہے کہ کسی مقصد کو حاصل کرنے کے لیے کون سے اقدامات کس ترتیب میں کرنے ہیں۔

سوال نمبر8: کمپیوٹیشنل تھنکنگ میں الگورتھم کا کردار کیا ہوتا ہے؟

جواب: کمپیوٹیشنل تھنکنگ میں الگورتھم حتمی حل کی نمائندگی کرتا ہے۔ تجزیہ، پیٹرن کی شناخت، اور تجرید کے مراحل سے گزرنے کے بعد، ہم جو مرحلہ وار منصوبہ تیار کرتے ہیں وہی الگورتھم کہلاتا ہے۔ یہ اس بات کو یقینی بناتا ہے کہ مسئلے کا حل واضح، موثر اور قابل عمل ہو، جسے انسان یا کمپیوٹر آسانی سے سمجھ کر نافذ کر سکے۔

سوال نمبر9: مسئلہ کی تفہیم کمپیوٹیشنل تھنکنگ میں کس طرح مدد کرتی ہے؟

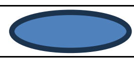
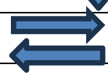
جواب: مسئلہ کی تفہیم (Problem Understanding) کمپیوٹیشنل تھنکنگ کا پہلا اور سب سے اہم اصول ہے۔ مسئلے کو اچھی طرح سمجھنے سے ہمیں اس کی بنیادی ضروریات، مقاصد اور حدود کا تعین کرنے میں مدد ملتی ہے۔ یہ ہمیں غیر متعلقہ تفصیلات سے بچاتا ہے اور اس بات کو یقینی بناتا ہے کہ ہم صحیح مسئلے کا صحیح حل تلاش کر رہے ہیں۔ واضح تفہیم کے بغیر، بنایا گیا حل غیر موثر یا غلط ہو سکتا ہے۔

سوال نمبر10: فلو چارٹ کیا ہیں اور وہ کس طرح استعمال ہوتے ہیں؟

جواب : فلو چارٹ (Flowchart) کسی عمل یا الگورتھم کے مراحل کی بصری نمائندگی ہوتے ہیں۔ وہ مختلف معیاری علامتوں کا استعمال کرتے ہیں جو مختلف قسم کے اقدامات (جیسے عمل، فیصلہ، ان پٹ / آؤٹ پٹ) کی نمائندگی کرتی ہیں، اور ان علامتوں کو تیروں کے ذریعے جوڑا جاتا ہے جو عمل کے بہاؤ کی سمت دکھاتے ہیں۔ یہ الگورتھم کے منطقی بہاؤ کو سمجھنے، ڈیزائن کرنے اور دوسروں کو سمجھانے کے لیے استعمال ہوتے ہیں۔

سوال نمبر 11: فلو چارٹ کی مختلف علامتیں کیا ہیں؟

جواب : فلو چارٹ میں استعمال ہونے والی چند اہم علامتیں یہ ہیں:

• **بیضوی (Oval)**	عمل کے آغاز یا اختتام (Start/End) کو ظاہر کرتی ہے۔	
• **مستطیل (Rectangle)**	کسی عمل یا پراسیس (Process) کو ظاہر کرتی ہے۔	
• **متوازی الاضلاع (Parallelogram)**	ڈیٹا کے ان پٹ یا آؤٹ پٹ (Input/Output) کو ظاہر کرتی ہے۔	
• **ہیرا (Diamond)**	کسی فیصلے یا شرط (Decision) کو ظاہر کرتی ہے جہاں بہاؤ دو یا زیادہ شاخوں میں تقسیم ہو سکتا ہے۔	
• **تیر (Arrow)**	عمل کے بہاؤ کی سمت (Flowline) کو ظاہر کرتا ہے۔	

سوال نمبر 12: فلو چارٹ میں پیرالوگرام اور مستطیل کا کیا مقصد ہوتا ہے؟

جواب:

- **مستطیل (Rectangle):** اس کا مقصد کسی بھی قسم کے عمل یا حساب کتاب (processing) کو ظاہر کرنا ہے، جیسے Sum = A + B۔
- **متوازی الاضلاع (Parallelogram):** اس کا مقصد ڈیٹا کو ان پٹ کرنے (جیسے صارف سے قدر لینا) یا آؤٹ پٹ کرنے (جیسے نتیجہ اسکرین پر دکھانا) کے عمل کو ظاہر کرنا ہے۔

سوال نمبر 13: فلو چارٹ کو شروع اور ختم کرنے کے لیے کون سی علامت استعمال ہوتی ہے؟

جواب : فلو چارٹ کو شروع (Start) اور ختم (End) کرنے کے لیے **بیضوی (Oval)** علامت استعمال ہوتی ہے۔

سوال نمبر 14: سوڈو کوڈ کیا ہوتا ہے؟

جواب : سوڈو کوڈ (Pseudocode) کسی الگورتھم کو بیان کرنے کا ایک غیر رسمی اور سادہ زبان میں لکھا گیا طریقہ ہے۔ یہ پروگرامنگ زبانوں کی ساخت اور عام انگریزی کی پڑھنے کی اہلیت کو یکجا کرتا ہے۔ یہ اصل کوڈ نہیں ہوتا اور اسے کمپیوٹر پر چلایا نہیں جا سکتا، لیکن یہ پروگرامرز کو کسی مخصوص پروگرامنگ زبان کے سنٹیکس کی فکر کیے بغیر الگورتھم کی منطق پر توجہ مرکوز کرنے میں مدد دیتا ہے۔

سوال نمبر 15: سوڈو کوڈ کے مقصد کی وضاحت کریں۔

جواب : سوڈو کوڈ کا بنیادی مقصد الگورتھم کی منطق کی منصوبہ بندی کرنا اور اسے واضح طور پر بیان کرنا ہے۔ یہ پروگرامرز کو اصل کوڈ لکھنے سے پہلے اپنے خیالات کو منظم کرنے میں مدد دیتا ہے۔ اس کے علاوہ، یہ ٹیم کے اراکین کے درمیان الگورتھم پر بات چیت کرنے کا ایک مؤثر ذریعہ ہے، چاہے وہ مختلف پروگرامنگ زبانیں ہی کیوں نہ استعمال کرتے ہوں۔

سوال نمبر 16: آپ فلو چارٹ اور سوڈو کوڈ کے درمیان فرق کیسے کرتے ہیں؟

جواب:

خصوصیت	سوڈو کوڈ (Pseudocode)	فلو چارٹ (Flowchart)
نمائندگی	متنی (سادہ زبان اور پروگرامنگ جیسی ساخت)	بصری / گرافیکل (علامتوں اور تیروں کا استعمال)
فارمیٹ	غیر رسمی اور لچکدار فارمیٹ	معیاری علامتوں پر مبنی سخت فارمیٹ
تفہیم	مرحلہ وار منطق کو تفصیل سے سمجھنا آسان	مجموعی بہاؤ کو ایک نظر میں سمجھنا آسان
تبدیلی	کوڈ میں تبدیل کرنا بہت آسان	کوڈ میں تبدیل کرنا نسبتاً مشکل

سوال نمبر 17: ٹائم کمپلیکسٹی سے آپ کی کیا مراد ہے؟

جواب : ٹائم کمپلیکسٹی (Time Complexity) اس بات کا پیمانہ ہے کہ ایک الگورتھم کتنا تیز یا سست کام کرتا ہے۔ یہ ظاہر کرتا ہے کہ ان پٹ کے سائز میں اضافے کے ساتھ الگورتھم کے چلنے کا وقت کس طرح تبدیل ہوتا ہے۔ اس کا استعمال مختلف الگورتھمز کا موازنہ کرنے کے لیے کیا جاتا ہے تاکہ یہ معلوم کیا جا سکے کہ کون سا زیادہ موثر ہے۔

سوال نمبر 18: سپیس کمپلیکسٹی سے کیا مراد ہے؟

جواب: سپیس کمپلیکسٹی (Space Complexity) اس بات کا پیمانہ ہے کہ ایک الگورتھم ان پٹ سائز کے لحاظ سے کتنی میموری (جگہ) استعمال کرتا ہے۔ اس میں ان پٹ کے لیے درکار میموری اور الگورتھم کے ذریعے استعمال ہونے والی کوئی بھی اضافی میموری دونوں شامل ہیں۔

سوال نمبر19: ڈرائی رن کیا ہے اور یہ کیوں ضروری ہے؟

جواب: ڈرائی رن :(Dry Run) یہ کسی الگورتھم یا فلو چارٹ کو نمونہ ڈیٹا کے ساتھ دستی طور پر مرحلہ وار چلانے کا عمل ہے تاکہ اس کے کام کرنے کے طریقے کو سمجھا جا سکے اور غلطیوں کی نشاندہی کی جا سکے۔ **اہمیت :** یہ اس لیے ضروری ہے کیونکہ یہ کمپیوٹر پر کوڈ چلانے سے پہلے ہی منطقی غلطیوں کو پکڑنے میں مدد کرتا ہے، جس سے وقت اور محنت کی بچت ہوتی ہے۔ یہ الگورتھم کے بہاؤ اور ہر مرحلے پر متغیرات کی قدروں کو سمجھنے میں بھی مدد کرتا ہے۔

سوال نمبر20: سیمولیشن کیا ہے؟ اس کے استعمالات لکھیں۔

جواب : سیمولیشن :(Simulation) یہ کسی حقیقی دنیا کے عمل یا نظام کا کمپیوٹر پروگرام کا استعمال کرتے ہوئے ایک ماڈل بنانے کا عمل ہے۔ **استعمالات:**

1. **الگورتھم کی جانچ :** یہ دیکھنے کے لیے کہ ایک الگورتھم مختلف قسم کے ڈیٹا کے ساتھ کیسا کام کرتا ہے۔
2. **مختلف منظرناموں کی کھوج :** مختلف حالات پیدا کر کے یہ دیکھنا کہ نظام پر کیا اثرات مرتب ہوتے ہیں، جیسے موسم کی پیشن گوئی یا ٹریفک کے بہاؤ کا مطالعہ۔

سوال نمبر 21 LARP : کیا ہے اور الگورتھم میں اس کی اہمیت کی وضاحت کریں۔

جواب :LARP: اس کا مطلب "Logic of Algorithms for Resolution of Problems" ہے۔ یہ ایک سافٹ ویئر ٹول ہے جو الگورتھم اور فلو چارٹ بنانے، چلانے اور سیکھنے کا ایک انٹرایکٹو طریقہ فراہم کرتا ہے۔ **اہمیت** LARP : کی اہمیت یہ ہے کہ یہ طالب علموں کو الگورتھم کے کام کرنے کے طریقے کو عملی طور پر دیکھنے اور سمجھنے میں مدد دیتا ہے۔ یہ مختلف ان پٹس کے آؤٹ پٹ پر اثرات کو ظاہر کرتا ہے اور الگورتھم لکھنے اور بہتر بنانے کی مشق کرنے کا ایک بہترین پلیٹ فارم ہے۔

سوال نمبر22: بگس اور اس کی اقسام کیا ہوتی ہیں؟

جواب : بگس :(Bugs) الگورتھم یا پروگرام میں موجود غلطیوں کو بگس کہا جاتا ہے جو اسے صحیح طریقے سے کام کرنے سے روکتی ہیں۔ **اقسام:**

1. **سنٹیکس کی غلطیاں :(Syntax Errors)** جب الگورتھم یا کوڈ کے لکھنے میں کوئی غلطی ہو، جیسے کوئی مرحلہ چھوٹ جانا یا غلط علامت کا استعمال۔
2. **رن ٹائم کی غلطیاں :(Runtime Errors)** یہ غلطیاں الگورتھم کے چلنے کے دوران ہوتی ہیں، جیسے صفر سے تقسیم کرنے کی کوشش کرنا۔
3. **منطقی غلطیاں :(Logical Errors)** یہ الگورتھم کی منطق میں غلطیاں ہوتی ہیں جس کی وجہ سے وہ چلتا تو ہے لیکن غلط نتیجہ دیتا ہے۔

سوال نمبر23: ڈیبگنگ کیا ہوتی ہے؟

جواب: ڈیبگنگ (Debugging) کسی الگورتھم، فلو چارٹ یا پروگرام میں موجود غلطیوں (بگس) کو تلاش کرنے اور انہیں ٹھیک کرنے کا عمل ہے۔

سوال نمبر24: دو ڈیبگنگ تکنیکس اور اس کی وضاحت کریں۔

جواب : دو عام ڈیبگنگ تکنیکیں درج ذیل ہیں:

1. **مراحل کو ٹریس کرنا :(Trace the Steps)** اس تکنیک میں، آپ اپنے الگورتھم یا فلو چارٹ کے ہر مرحلے سے دستی طور پر گزرتے ہیں تاکہ یہ دیکھ سکیں کہ غلطی کہاں ہو رہی ہے۔ ہر قدم پر متغیرات کی قدروں کو لکھنا اس عمل میں مددگار ثابت ہوتا ہے۔
2. **مسئلے کو سادہ کرنا :(Simplify the Problem)** اس تکنیک میں، آپ الگورتھم کو چھوٹے حصوں میں تقسیم کرتے ہیں اور ہر حصے کو الگ الگ جانچتے ہیں۔ اس سے یہ شناخت کرنا آسان ہو جاتا ہے کہ الگورتھم کا کون سا حصہ غلطی کا باعث بن رہا ہے۔

مشقی سوالات

سوال نمبر1: کمپیوٹیشنل تھنکنگ کی وضاحت کریں۔

جواب: کمپیوٹیشنل تھنکنگ (Computational Thinking) ایک مسئلہ حل کرنے کا عمل ہے جس میں ایسے طریقے اور تکنیک شامل ہیں جو کمپیوٹر سائنس کے اصولوں سے مطابقت رکھتے ہیں۔ یہ صرف کمپیوٹر سائنسدانوں کے لیے نہیں، بلکہ ہر اس شخص کے لیے ایک ضروری مہارت ہے جو پیچیدہ مسائل کو منظم اور منطقی طریقے سے حل کرنا چاہتا ہے۔ اس میں مسائل کو چھوٹے حصوں میں تقسیم کرنا، پیٹرن تلاش کرنا، غیر ضروری تفصیلات کو نظر انداز کرنا، اور مرحلہ وار حل تیار کرنا شامل ہے۔

سوال نمبر 2: کمپیوٹیشنل تھنکنگ میں decomposition کیا ہے؟

جواب: تجزیہ (Decomposition) کمپیوٹیشنل تھنکنگ کا وہ جزو ہے جس میں ایک بڑے اور پیچیدہ مسئلے کو چھوٹے، زیادہ قابل انتظام اور آسان کاموں میں تقسیم کیا جاتا ہے۔ اس سے ہر چھوٹے حصے پر الگ الگ توجہ مرکوز کرنا اور اسے حل کرنا آسان ہو جاتا ہے، جس سے مجموعی طور پر بڑا مسئلہ حل ہو جاتا ہے۔

سوال نمبر 3 Pattern recognition: کو ایک مثال کے ساتھ بیان کریں۔

جواب: پیٹرن کی شناخت (Pattern Recognition) وہ عمل ہے جس میں ڈیٹا یا مسائل کے اندر مماثلت، تکرار یا رجحانات کو تلاش کیا جاتا ہے۔ ان پیٹرنز کو پہچاننے سے ہمیں مسئلے کو زیادہ مؤثر طریقے سے حل کرنے میں مدد ملتی ہے۔ **مثال:** مربعوں کے رقبے میں ایک پیٹرن ہوتا ہے۔ اگر ایک مربع کی سائیڈ کی لمبائی 1,2,3,4 ہے، تو اس کا رقبہ 1,4,9,16 ہو گا۔ یہاں پیٹرن یہ ہے کہ ہر مربع کا رقبہ اس کی سائیڈ کی لمبائی کا مربع ($side\ length^2$) ہے۔

سوال نمبر 4: مسئلہ حل کرنے میں تجرید اور اس کی اہمیت کی وضاحت کریں۔

جواب: تجرید :(Abstraction) یہ وہ عمل ہے جس میں پیچیدہ نظاموں کو آسان بنانے کے لیے غیر ضروری تفصیلات کو نظر انداز کرتے ہوئے صرف ضروری اور متعلقہ معلومات پر توجہ مرکوز کی جاتی ہے۔ **اہمیت:** تجرید کی اہمیت یہ ہے کہ یہ ہمیں پیچیدگی کو کم کرنے میں مدد دیتی ہے۔ اس سے ہم مسئلے کی بنیادی ساخت کو سمجھ سکتے ہیں اور غیر ضروری تفصیلات میں الجھے بغیر ایک اعلیٰ سطحی حل تیار کر سکتے ہیں۔

سوال نمبر 5: الگورتھم کیا ہے؟

جواب: الگورتھم کسی مسئلے کو حل کرنے یا کسی کام کو مکمل کرنے کے لیے مرحلہ وار ہدایات کا ایک واضح اور منظم مجموعہ ہے۔ یہ ایک ترکیب کی طرح ہوتا ہے جو بتاتا ہے کہ کسی مقصد کو حاصل کرنے کے لیے کون سے اقدامات کس ترتیب میں کرنے ہیں۔

سوال نمبر 6: مسئلہ کی تفہیم کمپیوٹیشنل تھنکنگ میں کس طرح مدد کرتی ہے؟

جواب: مسئلہ کی تفہیم (Problem Understanding) کمپیوٹیشنل تھنکنگ کا پہلا اور سب سے اہم اصول ہے۔ مسئلے کو اچھی طرح سمجھنے سے ہمیں اس کی بنیادی ضروریات، مقاصد اور حدود کا تعین کرنے میں مدد ملتی ہے۔ یہ ہمیں غیر متعلقہ تفصیلات سے بچاتا ہے اور اس بات کو یقینی بناتا ہے کہ ہم صحیح مسئلے کا صحیح حل تلاش کر رہے ہیں۔

سوال نمبر 7: فلو چارٹ کیا ہیں اور وہ کس طرح استعمال ہوتے ہیں؟

جواب: فلو چارٹ (Flowchart) کسی عمل یا الگورتھم کے مراحل کی بصری نمائندگی ہوتے ہیں۔ وہ مختلف معیاری علامتوں کا استعمال کرتے ہیں جو مختلف قسم کے اقدامات کی نمائندگی کرتی ہیں، اور ان علامتوں کو تیروں کے ذریعے جوڑا جاتا ہے جو عمل کے بہاؤ کی سمت دکھاتے ہیں۔ یہ الگورتھم کے منطقی بہاؤ کو سمجھنے، ڈیزائن کرنے اور دوسروں کو سمجھانے کے لیے استعمال ہوتے ہیں۔

سوال نمبر 8: سوڈو کوڈ کے مقصد کی وضاحت کریں۔

جواب: سوڈو کوڈ کا بنیادی مقصد الگورتھم کی منطق کی منصوبہ بندی کرنا اور اسے واضح طور پر بیان کرنا ہے۔ یہ پروگرامرز کو اصل کوڈ لکھنے سے پہلے اپنے خیالات کو منظم کرنے میں مدد دیتا ہے اور ٹیم کے اراکین کے درمیان الگورتھم پر بات چیت کرنے کا ایک مؤثر ذریعہ ہے۔

سوال نمبر 9: آپ فلو چارٹ اور سوڈو کوڈ کے درمیان فرق کیسے کرتے ہیں؟

جواب: فلو چارٹ ایک بصری اور گرافیکل نمائندگی ہے جو علامتوں کا استعمال کرتی ہے، جبکہ سوڈو کوڈ ایک متنی نمائندگی ہے جو سادہ زبان کا استعمال کرتی ہے۔ فلو چارٹ مجموعی بہاؤ کو سمجھنے کے لیے بہتر ہے، جبکہ سوڈو کوڈ مرحلہ وار منطق کی تفصیل کے لیے بہتر ہے اور اسے کوڈ میں تبدیل کرنا آسان ہوتا ہے۔

سوال نمبر 10: ڈرائی رن کیا ہے اور یہ کیوں ضروری ہے؟

جواب :ڈرائی رن :(Dry Run) یہ کسی الگورتھم کو نمونہ ڈیٹا کے ساتھ دستی طور پر مرحلہ وار چلانے کا عمل ہے۔ **اہمیت :** یہ اس لیے ضروری ہے کیونکہ یہ کوڈ چلانے سے پہلے ہی منطقی غلطیوں کو پکڑنے میں مدد کرتا ہے، جس سے وقت اور محنت کی بچت ہوتی ہے اور الگورتھم کی درستگی کو یقینی بنایا جاتا ہے۔

سوال نمبر11 LARP : کیاہے اور الگورتھم میں اس کی اہمیت کی وضاحت کریں۔

جواب :LARP : اس کا مطلب "Logic of Algorithms for Resolution of Problems" ہے۔ یہ ایک سافٹ ویئر ٹول ہے جو الگورتھم اور فلو چارٹ بنانے اور چلانے کا ایک انٹرایکٹو طریقہ فراہم کرتا ہے۔ **اہمیت** LARP : کی اہمیت یہ ہے کہ یہ طالب علموں کو الگورتھم کے کام کرنے کے طریقے کو عملی طور پر دیکھنے اور سمجھنے میں مدد دیتا ہے، جس سے سیکھنے کا عمل آسان اور دلچسپ ہو جاتا ہے۔

سوال نمبر12: دو ڈیبگنگ تکنیکس اور اس کی وضاحت کریں۔

جواب : دو عام ڈیبگنگ تکنیکیں درج ذیل ہیں:

1. **مراحل کو ٹریس کرنا :(Trace the Steps)** اس تکنیک میں، آپ اپنے الگورتھم کے ہر مرحلے سے دستی طور پر گزرتے ہیں تاکہ یہ دیکھ سکیں کہ غلطی کہاں ہو رہی ہے۔
2. **مسئلے کو سادہ کرنا :(Simplify the Problem)** اس تکنیک میں، آپ الگورتھم کو چھوٹے حصوں میں تقسیم کرتے ہیں اور ہر حصے کو الگ الگ جانچتے ہیں تاکہ غلطی کے ماخذ کی شناخت کی جا سکے۔

اہم ترین انشائیہ سوالات

یونٹ 7: کمپیوٹیشنل تھنکنگ

کمپیوٹیشنل تھنکنگ میں تقسیم (decomposition) کے تصور پر تبادلہ خیال کریں۔ یہ کیوں اہم ہے؟
کمپیوٹیشنل تھنکنگ کے حوالے سے پیٹرن کی شناخت کی وضاحت کریں۔ یہ مسئلہ حل کرنے میں کس طرح مدد کرتی ہے؟
کمپیوٹیشنل تھنکنگ میں abstraction کیا ہے؟ اس کی اہمیت پر تبادلہ خیال کریں اور مثالیں فراہم کریں کہ پیچیدہ مسائل کو آسان بنانے کے لیے کس طرح اس کا استعمال کیا جا سکتا ہے؟
LARP کیا ہے؟ الگورتھم سیکھنے اور مشق کرنے میں اس کی اہمیت پر تبادلہ خیال کریں۔