LEVEL 1 TEST 3 PROGRAMMING SET D

Study the following program written in a block-structured language:

#include

int x, y;

void P(int n) {

} void Q() {

int x, y;

printf("%d\\n", x); // Assuming printf is used for writing to the console in C

int main() {

x=7;

printf("%d\\n", x);

return 0;

What will be printed by write statements marked(l) and (2) in the program if variables are statically scoped?

3,6

6,7

3,7

None of these

The following C declarations:

struct node

int i;

float j;

struct node •s[10] ;

define s to be

An array, each element of which is a pointer to a structure of type node

A structure of 2 fields, each field being a pointer to an array of 10 elements

A structure of 3 fields: an integer, a float, and an array of 10 elements

An array, each element of which is a structure of type node.

The value of j at the end of the execution of the following C program

```
intincr (int i)
staticint count = 0;
count = count + i;
return (count);
main () {
int i, j;
for (i = O; i 4;
j = incr (i);
```

10

4

6

7

The following C function takes two ASCII strings and determines whether one is an anagram of the other.

An anagram of a string s is a string obtained by permuting the letters in s.

```
int anagram (char *a, char *b)
int count [128], j;
for (j = 0; j < 128; j++) count[j] = 0;
while (affl bj]) {
B;
for (j = 0; j < 128; j++) if (count 01) return O;
return 1;
}
```

Choose the correct alternative for statements A and B

A : count [a[j]]++ and B : count[b0]]—

A: count [a[j]]++ and B : count[b0]]++

A : count and B :

A : count [a[jl]++and B : count[b0++]]—

Which of the following library function is used to concatenate two strings in a C program?

strcpyf()

strlwr()

strcomp()

Strcat()

strcmp()

Which of the following functions is used to copy one memory location to another in C++?

memmove

memchr

memcmp

memcpy

memset

What would happen if "String[]args" is not included as an argument in the main method?

No error

Compilation error

The program won't run

Program exit

Which of the following do not represent legal flow control statements?

break;

return;

exit();

continue outer;

Where is an array stored in memory?

heap space

stack space

heap space and stack space

first generation memory

To turn off a particular bit in a number, which of the following operators is suitable?

!

&

||

&&

To access the 9th element of an array "arr", which of the following statements is used?

arr[9]

arr[8]

arr(8)

arr(9)

What will be the content of the array "A" after the following operation:

int] =

[gv: garbage value]

0 gv gv gv gv

gv gv gv gv 0

gv gv gv gv gv

00000

Which operator among the following has the highest precedence?

Unary

==

Postfix

<<

To turn on a particular bit in a number, which of the following operators is suitable?

&

|

||

&&

To access the nth element of an array "arr", which of the following statements is used?

arr(n - 1)

arr[n]

arr[n- 1]

arr(n - 1)

What will be the content of the array "A" after the following operation:

intA[5] = {7};

[gv: garbage value]

7 gv gv gv gv

gv gv gv gv 7

77777

00000

int = 2, 3}, {4, 5, 6}, {7, S, 9}};

mt *ptr =

printf("%d" *(ptr+

What will be the output?

5

6

7

8

What will be the output of the code?

```
#include <iostream>

Int malno {

int x = 1 0;

int* ptr =

int** ptr_ptr = &pW•,

std::cout << x << std: :€ndl,

retum 0,
```

10

20

O (Zero)

Compilation Error

How do you define a function that returns a reference in C++?

dataType& functionName(parameters)

&dataType functionName(parameters)

dataType functionName&(parameters)

dataType functionName(parameters)&

A C++ program utilizes a complex data structure with nested pointers and dynamic memory allocations. During program execution, memory access violations occur, leading to segmentation faults. What could be a potential reason for these memory access violations?

The program exceeds the maximum stack size allocated for dynamic memory.

Memory leaks occur, resulting in excessive memory consumption

The program accesses deallocated memory locations

The custom memory allocator incorrectly handles memory alignment requirements

What is the primary characteristic of problems that are suitable for backtracking algorithms?

Optimal substructure

Greedy-choice property

Overlapping subproblems

Exhaustive search space

What is the top down approach in dynamic programming?

A way to store and organise data in a computer program

A technique for breaking down a problem into smaller subproblems

A process of optimising memory usage in a program

A method of solving problems by starting with overall problems and recursively breaking it down into smaller subproblems

Which of the following problems should be solved using dynamic programming?

Merge Sort

Binary Search

Longest common subsequence

Quick Sort

Backtracking algorithm is implemented by constructing a tree of choices called as?

State-space tree

State-chart tree

Node tree

Backtracking tree

Time complexity of fractional knapsack problem is

O(n log n)

O(n)

0(n2)

0(nW)

In linear hashing, formula used to calculate number of records if blocking factor, loading factor and file buckets are known is as

r =l + bfr + N

r =l - bfr - N

r= l + bfr- N

r =l * bfr * N

To insert a new node at the beginning of a linked list, what is the time complexity of the operation?

0(1)

O(log n)

O(n)

0(n^2)

Consider the following pseudo code. Assume that IntQueue is an integer queue. What does the function fun do?

```
void fun(int n)
IntQueue q = new IntQueue();
q.enqueue(O);
q.enqueue(l );
for (int i O; i < n;
int a = q-dequeue();
int b = q-dequeue();
q.enqueue(b);
q.enqueue(a + b);
print(a);
```

Prints numbers from 0 to n-1

Prints numbers from n-1 to O

Prints first n Fibonacci numbers

Prints first n Fibonacci numbers in reverse order

Fixed-length strings programmers specify the length while declaring the string.

TRUE

FALSE

Can be true or false

Can not say

What is the maximum height of an AVL tree with p nodes?

P

log(p)

log(p)/2

P/2