**General Instructions:**

* **Follow the instructions given in each section.**
* **Make sure that you attempt the questions in order.**

**SECTION-A (10\*1 mark=10 marks)**

***(All questions are compulsory)***

Q1 What is the time complexity of a recursive function that calls itself n times?

a) O(n)

b) O(log n)

c) O(n log n)

d) O(n^2)

Answer: a) O(n)

Q2 Which of the following is true about recursion in C programming?

a) It can be used to solve problems that are difficult to solve using iteration

b) It can be used to solve all problems

c) It is always faster than iteration

d) None of the above

Answer: a) It can be used to solve problems that are difficult to solve using iteration

Q3 Which of the following is not required in a recursive function?

a) Base case

b) Recursive call

c) Loop

d) All of the above are required

Answer: c) Loop

Q4 1. Which operator is used to access members of a union in C?

A) .

B) ->

C) &

D) \*

Answer: A) .

Q5 1. Which of the following is an example of a typedef declaration for a struct in C?

A) typedef int my\_int;

B) typedef struct { int x; int y; } point;

C) typedef float my\_float;

D) None of the above

Answer: B) typedef struct { int x; int y; } point;

Q6 1. What is the keyword used to define an enumeration in C?

A) enum

B) define

C) typedef

D) struct

Answer: A) enum

Q7 1. What is the syntax for declaring a typedef for a union in C?

a. typedef union { } union\_name;

b. typedef struct { } struct\_name;

c. typedef enum { } enum\_name;

d. None of the above

Answer: a

Q8 1. 1. What is the purpose of a union in C?

a. To save memory by allowing multiple data types to use the same memory location

b. To group related data items together

c. To define a new data type that can hold multiple values

d. None of the above

Answer: a

Q9 1. What is the return value of the fwrite() function if an error occurs?

a) -1

b) 0

c) 1

d) NULL

Answer: b) 0

Q10 1. What is the purpose of the feof() function in C file handling?

a) to check if the end of the file has been reached

b) to check if an error occurred while reading the file

c) to check if the file is open

d) to check if the file is writable

Answer: a) to check if the end of the file has been reached

**SECTION-B (5\*2 mark=10 marks)**

***(All questions are compulsory)***

Q11 What is the output of the following recursive function when called with an argument of 3?

void printSeries(int n) {

if (n == 0) {

return;

}

printSeries(n-1);

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

}

A. 1 4 9

B. 9 4 1

C. 1 9 4

D. 4 1 9

Correct answer: A (1 4 9)

Q12 What is the output of the following code?

#include <stdio.h>

struct person {

char name[20];

int age;

};

void printPerson(struct person \*p) {

printf("%s %d\n", p->name, p->age);

}

int main() {

struct person p1 = { "John", 25 };

struct person \*p2 = &p1;

printPerson(p2);

return 0;

}

A. John 25

B. John

C. 25

D. Error

Correct answer: A

Q13 Which of the following is an example of accessing a member of a structure variable called s using the . operator?

a. s->id

b. s.id

c. \*s.id

d. &s.id

Correct answer: b

Q14 What is the output of the following code snippet?

#include <stdio.h>

enum status { FAIL, PASS };

int main() {

enum status s1 = FAIL, s2 = PASS;

printf("%d %d", s1, s2);

return 0;

}

a) 0 1

b) 1 0

c) Compiler error

d) Runtime error

Answer: a

Q15 What is the output of the following program?

#include <stdio.h>

union data {

int i;

float f;

char c[4];

};

int main() {

union data d = {.c = "abc"};

printf("d.i = %d\n", d.i);

printf("d.f = %f\n", d.f);

printf("d.c = %s\n", d.c);

return 0;

}

a) d.f = 0.000000, d.c = abc

b) d.i = garbage value, d.f = garbage value, d.c = abc

c) Compiler error

d) None of the above

Correct answer: a

**SECTION-C(Coding Question) (2x5 marks=5 marks)**

Q16 You are currently standing on the left and topmost cell of the N\*M matrix you have to go to the right and bottom most cells. Count all the possible paths from top left to bottom right of a MxN matrix with the constraints that from each cell you can either move to right or down.

**Input:**

Enter integer M and integer N

**Constraints:**

1 ≤ M, N ≤ 10

**Output:**

**print the number of paths..**

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 3 3  Let the given input 3\*3 matrix is filled  as such:  A B C  D E F  G H I  The possible paths which exists to reach  'I' from 'A' following above conditions  are as follows:ABCFI, ABEHI, ADGHI, ADEFI,  ADEHI, ABEFI | 6 |
| STC2 | 4 6 | 56 |

**Solution 16:**

#include <stdio.h>

long long func(int i,int j)

{

if(i==0 && j==0) return 1;

if(i<0 || j<0 ) return 0;

int up=func(i-1,j);

int left=func(i,j-1);

return up +left;

}

long long numberOfPaths(int m, int n)

{

// Code Here

return func(m-1,n-1);

}

int main()

{

long long N,R;

scanf("%ld%ld",&N,&R);

long long ans =numberOfPaths(N,R);

printf("%ld",ans);

}

Test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| TC1 | 7 9 | 3003 |
| TC2 | 4 5 | 35 |
| TC3 | 3 6 | 21 |
| TC4 | 3 4 | 10 |
| TC5 | 4 6 | 56 |

Q17  **Write a C program to find the greatest common divisor(GCD) of two numbers using recursion.**

**For a set of positive integers (a, b), the greatest common divisor is defined as the greatest positive number which is a common factor of both the positive integers (a, b).**

**Input :**

Enter an integer n.

**Constraints:**

1<=**n**<=100

**Output:**

**print a integer gcd of n;**

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 2 5  Explanation:The highest common factor is 1. | 1 |
| STC2 | 8 2  Explanation:The highest common factor is 2. | 2 |

**Solution 17:**

#include <stdio.h>

int gcd(int a, int b);

int main() {

int a, b;

scanf("%d%d", &a, &b);

printf("%d",gcd(a, b));

return 0;

}

int gcd(int a, int b) {

if (b == 0) {

return a;

} else {

return gcd(b, a % b);

}

}

Test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| TC1 | 33 55 | 11 |
| TC2 | 100 585 | 5 |
| TC3 | 999 333 | 3 |
| TC4 | 198 24 | 6 |
| TC5 | 17 25 | 1 |

**SECTION-D (Coding Question)(1x10 mark=10 mark)**

Q18 **Rishabh wants to check file content ,so help him to compare the contents of two files using C Program.**

**Sample Input**:

files named file1.txt and file2.txt where file2.txt does not exist in the directory

**Sample Output**:

file2.txt File can not be opened :

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Case 1 | Test Case 2 | Test Case 3 |
| Input | files named file1.txt and file2.txt where file2.txt does not exist in the directory | iles named file1.txt and file2.txt with different contents "Hello World!" and "Goodbye World!" | iles named file1.txt and file2.txt with the same contents "Hello World!" |
| Output | file2.txt File can not be opened : | File contents are not same. | File contents are same. |

**Solution:**

#include <stdio.h>

#include <string.h>

int main(int argc, char \*argv[])

{

FILE \*fp1 ;

FILE \*fp2 ;

int cnt1 = 0;

int cnt2 = 0;

int flg = 0;

if( argc < 3 )

{

printf("Insufficient Arguments!!!\n");

printf("Please use \"program-name file-name1 file-name2\" format.\n");

return -1;

}

fp1 = fopen(argv[1],"r");

if( fp1 == NULL )

{

printf("\n%s File can not be opened : \n",argv[1]);

return -1;

}

// move file pointer to end and get total number of bytes

fseek(fp1,0,SEEK\_END);

cnt1 = ftell(fp1);

fp2 = fopen(argv[2],"r");

if( fp2 == NULL )

{

printf("\n%s File can not be opened : \n",argv[2]);

return -1;

}

// move file pointer to end and get total number of bytes

fseek(fp2,0,SEEK\_END);

cnt2 = ftell(fp2);

fseek(fp1,0,SEEK\_SET);

fseek(fp2,0,SEEK\_SET);

// check for the total number of bytes

if( cnt1 != cnt2 ){

printf("\nFile contents are not same\n");

}

else

{

while( ! feof(fp1) )

{

if( fgetc(fp1) != fgetc(fp2) )

{

flg = 1;

break;

}

}

if( flg ) printf("\nFile contents are not same.\n");

else printf("\nFile contents are same.\n");

}

fclose(fp1);

fclose(fp2);

return 0;

}