**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 function is used to write a single character to a file in C?

a) fprintf()

b) fputc()

c) fputs()

d) fread()

Answer: b) fputc()

Q2 What function is used to read a block of data from a file in C?

a) fread()

b) fgetc()

c) fgets()

d) fprintf()

Answer: a) fread()

Q3 What is the return value of the fopen() function if the file cannot be opened?

a) NULL

b) 0

c) -1

d) 1

Answer: a) NULL

Q4 Which of the following is not a valid operation on an enumeration in C?

a. Incrementing

b. Decrementing

c. Multiplying

d. None of the above

Answer: c

Q5 What is the purpose of a typedef in C?

a. To create a new data type with a different name

b. To define a new variable

c. To declare a new function

d. None of the above

Answer: a

Q6 Which keyword is used to declare a typedef in C?

a. typedef

b. struct

c. union

d. None of the above

Answer: a

Q7 Which of the following is an example of a union in C?

A) union { int x; float y; };

B) union { char a; char b; };

C) union { int x; char y; };

D) All of the above

Answer: D) All of the above

Q8 What is the size of a union in C?

A) The size of the largest member

B) The size of the smallest member

C) The sum of the sizes of all members

D) None of the above

Answer: A) The size of the largest member

Q9 What is the purpose of a helper function in recursion?

a) It is used to make the code shorter

b) It is used to make the code faster

c) It is used to avoid using global variables

d) None of the above

Answer: c) It is used to avoid using global variables

Q10 Which of the following is an example of a recursive function?

a) Factorial function

b) Fibonacci function

c) Tower of Hanoi function

d) All of the above

Answer: d) All of the above

**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 4?

void printBinary(int n) {

if (n == 0) {

return;

}

printBinary(n/2);

printf("%d", n%2);

}

A. 101

B. 111

C. 100

D. 110

Q12 What is the output of the following code?

#include <stdio.h>

struct student {

char name[20];

int age;

};

int main() {

struct student s1 = { "John", 20 };

struct student s2 = s1;

s2.age = 25;

printf("%s %d\n", s1.name, s1.age);

printf("%s %d\n", s2.name, s2.age);

return 0;

}

A. John 20 John 25

B. John 20 John 20

C. John 25 John 25

D. Error

Correct answer: A

Q13 How can you declare a variable of the student type after using typedef to define the structure?

a. student s;

b. struct student s;

c. int s;

d. char s[50];

Correct answer: a

Q14 What is the output of the following code snippet?

#include <stdio.h>

enum { A, B, C = 1, D, E };

int main() {

printf("%d", E);

return 0;

}

a) 2

b) 3

c) 4

d) 5

Answer: b

Q15 What is the output of the following program?

#include <stdio.h>

union data {

int i;

float f;

};

int main() {

union data d = {10};

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

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

return 0;

}

a) d.i = 10, d.f = 0.000000

b) d.i = 10, d.f = garbage value

c) Compiler error

d) None of the above

Correct answer: a

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

Q16.Given two integers n and r, find nCr. Since the answer may be very large, calculate the answer modulo 109+7.

**Input:**

**input n and r.**

**Constraints:**

**1 ≤ n ≤ 30**

**1 ≤ r ≤ 30**

char will contain upper and lower case alphabets.

**Output:**

nCr modulo 10^9 + 7

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 3 2  Explanation: 3C2 = 3. | 3 |
| STC2 | 2 4  Explanation: r is greater than n. | 0 |

**Solution 16:**

#include <stdio.h>

#define MOD 1000000007

int nCr(int n, int r) {

if (r == 0 || r == n)

return 1;

int ans = (nCr(n-1, r-1) % MOD + nCr(n-1, r) % MOD) % MOD;

return ans;

}

int main() {

int n,r;

scanf("%d%d",&n,&r);

int ans = nCr(n, r);

printf("%d",ans);

return 0;

}

Test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| TC1 | 12 20 | 0 |
| TC2 | 6 3 | 20 |
| TC3 | 20 12 | 125970 |
| TC4 | 19 5 | 11628 |
| TC5 | 29 21 | 4292145 |

Q17. You are a programmer who has been tasked with writing a C program to find the nth Fibonacci number using recursion. Your boss wants the program to be able to handle large values of n without crashing or slowing down too much. How would you approach this task?

**Input:**

**an integer n**

**Constraints:**

0<= n <= 30

**Output:**

**print nth term of**  Fibonacci sequence(0,1,1,2,3,5………)

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 0 | 0 |
| STC2 | 5 | 5 |

**Solution 17:**

#include <stdio.h>

int fibonacci(int num) {

if (num == 0 || num == 1) {

return num;

} else {

return fibonacci(num - 1) + fibonacci(num - 2);

}

}

int main() {

int num;

scanf("%d", &num);

printf("%d\n",fibonacci(num));

return 0;

}

Test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| TC1 | 10 | 5510 |
| TC2 | 15 | 610 |
| TC3 | 30 | 832040 |
| TC4 | 1 | 1 |
| TC5 | 0 | 0 |

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

Q18 **I work as a network administrator at a large corporation, and one of my responsibilities is to monitor the network traffic and identify any potential security threats. To do this, I need to know the class of each IP address that is connected to the network. Using a program to find the class of an IP address, I am able to quickly determine whether a particular address is a part of a private or public network, which helps me to take appropriate action in case of any security breach**

**Sample Input**:

Enter IP Address (xxx.xxx.xxx.xxx format): 192.168.1.1

**Sample Output**:

Ip Address: 192. 168. 1. 1 Class C Ip Address.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Case 1 | Test Case 2 | Test Case 3 |
| Input | Enter IP Address (xxx.xxx.xxx.xxx format): 192.168.1.1 | Enter IP Address (xxx.xxx.xxx.xxx format): 10.0.0.1 | Enter IP Address (xxx.xxx.xxx.xxx format): 172.31.255.255 |
| Output | Ip Address: 192. 168. 1. 1 Class C Ip Address. | Ip Address: 10.0.0.1 Class A Ip Address. | Ip Address: 172. 31. 255. 255 Class B Ip Address. |

**Solution:**

#include <stdio.h>

#include <string.h>

void extractIpAddress(unsigned char \*sourceString,short \*ipAddress)

{

unsigned short len=0;

unsigned char oct[4]={0},cnt=0,cnt1=0,i,buf[5];

len=strlen(sourceString);

for(i=0;i<len;i++)

{

if(sourceString[i]!='.'){

buf[cnt++] =sourceString[i];

}

if(sourceString[i]=='.' || i==len-1){

buf[cnt]='\0';

cnt=0;

oct[cnt1++]=atoi(buf);

}

}

ipAddress[0]=oct[0];

ipAddress[1]=oct[1];

ipAddress[2]=oct[2];

ipAddress[3]=oct[3];

}

int main()

{

unsigned char ip[20]={0};

short ipAddress[4];

printf("Enter IP Address (xxx.xxx.xxx.xxx format): ");

scanf("%s",ip);

extractIpAddress(ip,&ipAddress[0]);

printf("\nIp Address: %03d. %03d. %03d. %03d\n",ipAddress[0],ipAddress[1],ipAddress[2],ipAddress[3]);

if(ipAddress[0]>=0 && ipAddress[0]<=127)

printf("Class A Ip Address.\n");

if(ipAddress[0]>127 && ipAddress[0]<191)

printf("Class B Ip Address.\n");

if(ipAddress[0]>191 && ipAddress[0]<224)

printf("Class C Ip Address.\n");

if(ipAddress[0]>224 && ipAddress[0]<=239)

printf("Class D Ip Address.\n");

if(ipAddress[0]>239)

printf("Class E Ip Address.\n");

return 0;

}