**10 MCQ (1 mark each)**

Q.1. typedef's have the advantage that they obey scope rules, that is they can be declared local to a function or a block whereas #define's always have a global effect.

A) Yes

B) No

Q.2. Which function is used to close an opened file in C?

A) close()

B) fclose()

C) file\_close()

D) fileclose()

Q.3. A union cannot be nested in a structure.

A) True

B) False

Q.4. The size of a union is \_\_\_.

A) Sum of sizes of all members

B) Predefined by the compiler

C) Equal to size of largest data type

D) None of these

Q.5. Which is the correct syntax to create a union?

A) union union\_name {

};

B) union union\_name {

}

C) union union\_name (

);

D) Union union\_name (

)

Q.6. The elements of union are always accessed using & operator.

A) Yes

B) No

Q.7. A structure can be nested inside another structure.

A) True

B) False

Q.8. A structure can contain similar or dissimilar elements.

A) True

B) False

Q.9. Usually recursion works slower than loops.

A) Yes

B) No

Q.10. Which function is used to delete an existing file in C?

A) delete()

B) fremove()

C) frem()

D) remove()

**5 MCQ (2 mark each)**

Q.1. What will following code do?

#include<stdio.h>

int reverse(int);

int main()

{

int no=5;

reverse(no);

return 0;

}

int reverse(int no)

{

if(no == 0)

return 0;

else

printf("%d,", no);

reverse (no--);

}

A) Print 5, 4, 3, 2, 1

B) Print 1, 2, 3, 4, 5

C) Print 5, 4, 3, 2, 1, 0

D) Infinite loop

Q.2. What will be the output of the program?

#include<stdio.h>

int main()

{

union var

{

int a, b;

};

union var v;

v.a=10;

v.b=20;

printf("%d\n", v.a);

return 0;

}

A) 10

B) 20

C) 30

D) 0

Q.3. What will be the output of the program?

#include<stdio.h>

int main()

{

enum status {pass, fail, absent};

enum status stud1, stud2, stud3;

stud1 = pass;

stud2 = absent;

stud3 = fail;

printf("%d %d %d\n", stud1, stud2, stud3);

return 0;

}

A) 0, 1, 2

B) 1, 2, 3

C) 0, 2, 1

D) 1, 3, 2

Q.4. What will be the output of the program?

#include<stdio.h>

struct course

{

int courseno;

char coursename[25];

};

int main()

{

struct course c[] = { {102, "Java"},

{103, "PHP"},

{104, "DotNet"} };

printf("%d ", c[1].courseno);

printf("%s\n", (\*(c+2)).coursename);

return 0;

}

A) 103 DotNet

B) 102 Java

C) 103 PHP

D) 104 DotNet

Q.5. Point out the error in the program?

struct emp

{

int ecode;

struct emp e;

};

A) Error: in structure declaration

B) Linker Error

C) No Error

D) None of above

**2 Coding Questions (5 mark each)**

Q.1. C program to read information of single student. It contains Name, Roll number, Birthday, admission date. Calculate age of student at the time of admission with using structure.

**Sample Input 1**

101

Sagar

15-03-1998

15-04-2022

**Sample Output 1**

24 Years

**Sample Input 2**

103

Khushi

12-01-1996

12-03-2023

**Sample Output 2**

27 Years

**Input Explanation**

Input consists of four lines

First input consists integer value which represents roll number of student,

Second input consists string value which represents name of student

Third input consists date in (dd-mm-yyyy) format, separated by hyphen which represents date of birth of student

Fourth input consists date in (dd-mm-yyyy) format, separated by hyphen which represents date of admission of student

**Output Explanation**

Output consists of single integer value which represents present age of student in years.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 110  Jishan  03-04-1990  04-06-2012 | 121  Kavish  04-04-1991  04-06- 2022 | 132  Kiara  14-05-1992  12-10-2021 | 133  Rohit  16-05-1994  14-03-2022 | 126  Ram  12-09-1995  04-03-2022 |
| **Output** | 22 Years |  |  |  |  |

**#Solution**

#include<stdio.h>

struct student

{

int roll\_num;

char name[20];

struct Date

{

int D;

int M;

int Y;

}bd,ad;

};

void main()

{

int age;

struct student a;

scanf("%d",&a.roll\_num);

scanf("%s",a.name);

scanf("%d-%d-%d",&a.bd.D,&a.bd.M,&a.bd.Y);

scanf("%d-%d-%d",&a.ad.D,&a.ad.M,&a.ad.Y);

age=a.ad.Y-a.bd.Y;

printf("%d Years",age);

}

Q.2. Write a program in C to find the sum of digits of a given number using recursion.

**Sample Input 1**

25

**Sample Output 1**

7

**Sample Input 2**

256

**Sample Output 2**

13

**Input Explanation**

Input consists of single integer value

**Output Explanation**

Output consists of single integer value that is sum of digits of given number

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 456 | 123 | 485 | 159 | 753 |
| **Output** | 15 | 6 | 17 | 15 | 15 |

**#Solution**

#include <stdio.h>

int DigitSum(int num);

int main()

{

int n1, sum;

scanf("%d", &n1);

sum = DigitSum(n1);

printf("%d", sum);

return 0;

}

int DigitSum(int n1)

{

if(n1 == 0)

return 0;

return ((n1 % 10) + DigitSum(n1 / 10));

}

**1 Coding Question (10 mark)**

Q.1. Write a program in C to Print Fibonacci Series using recursion, Input number of terms for the Series.

**Sample Input 1**

10

**Sample Output 1**

1 1 2 3 5 8 13 21 34 55

**Sample Input 2**

5

**Sample Output 2**

1 1 2 3 5

**Input Explanation**

Input consists of single integer value n

**Output Explanation**

Output consists of multiple space separated integer values in Fibonacci sequence upto nth term.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 10 | 5 | 7 | 8 | 3 |
| **Output** | 1 1 2 3 5 8 13 21 34 55 | 1 1 2 3 5 | 1 1 2 3 5 8 13 | 1 1 2 3 5 8 13 21 | 1 1 2 |

**#Solution**

#include<stdio.h>

int term;

int fibonacci(int prNo, int num);

void main()

{

static int prNo = 0, num = 1;

scanf("%d", &term);

printf("1 ");

fibonacci(prNo, num);

}

int fibonacci(int prNo, int num)

{

static int i = 1;

int nxtNo;

if (i == term)

return (0);

else

{

nxtNo = prNo + num;

prNo = num;

num = nxtNo;

printf("%d ", nxtNo);

i++;

fibonacci(prNo, num);

}

return (0);

}