**10 MCQ (1 mark each)**

Q.1. Is it true that too many recursive calls may result into stack overflow?

A) Yes

B) No

Q.2. Which of the following statement is True?

A) User has to explicitly define the numeric value of enumerations

B) User has a control over the size of enumeration variables.

C) Enumeration can have an effect local to the block, if desired

D) Enumerations have a global effect throughout the file.

Q.3. Nested unions are allowed.

A) True

B) False

Q.4. In which case union is better than structure?

A) Less memory is available

B) Faster compilation is required

C) When functions are included

D) None of these

Q.5. Which of the following is the collection of different data types?

A) structure

B) string

C) array

D) All of the above

Q.6. Which keyword is used to define a union?

A) un

B) union

C) Union

D) None of these

Q.7. The '->' operator can be used to access structures elements using a pointer to a structure variable only.

A) True

B) False

Q.8. Which is the correct syntax to declare a file pointer in C?

A) File \*file\_pointer;

B) FILE \*file\_pointer;

C) File file\_pointer;

D) FILE \*file\_pointer;

Q.9. Which function is used to seek the file pointer position in C?

A) seek()

B) fseek()

C) fileseek()

D) fmove()

Q.10. Is the following declaration acceptable?

typedef long no, \*ptrtono;

no n;

ptrtono p;

A) Yes

B) NO

**5 MCQ (2 mark each)**

Q.1. What will be the output of the program?

#include<stdio.h>

int sumdig(int);

int main()

{

int a, b;

a = sumdig(123);

b = sumdig(123);

printf("%d, %d\n", a, b);

return 0;

}

int sumdig(int n)

{

int s, d;

if(n!=0)

{

d = n%10;

n = n/10;

s = d+sumdig(n);

}

else

return 0;

return s;

}

A) 4, 4

B) 3, 3

C) 6, 6

D) 12, 12

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

#include<stdio.h>

int main()

{

struct value

{

int bit1:1;

int bit3:4;

int bit4:4;

}bit={1, 2, 13};

printf("%d, %d, %d\n", bit.bit1, bit.bit3, bit.bit4);

return 0;

}

A) 1, 2, 13

B) 1, 4, 4

C) -1, 2, -3

D) -1, -2, -13

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

#include<stdio.h>

int main()

{

struct emp

{

char \*n;

int age;

};

struct emp e1 = {"Dravid", 23};

struct emp e2 = e1;

strupr(e2.n);

printf("%s\n", e1.n);

return 0;

}

A) Error: Invalid structure assignment

B) DRAVID

C) Dravid

D) No output

Q.4. 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

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

#include<stdio.h>

int main()

{

struct emp

{

char name[20];

float sal;

};

struct emp e[10];

int i;

for(i=0; i<=9; i++)

scanf("%s %f", e[i].name, &e[i].sal);

return 0;

}

A) Error: invalid structure member

B) Error: Floating point formats not linked

C) No error

D) None of above

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

Q.1. A Company wants to organize the data of their employees. Help them with a C program to accept the details of employee and display them using structure. Details consist of Employee ID, Name, Designation, Department, Salary.

**Sample Input 1**

1

Rahul

Trainer

IT

60000

**Sample Output 1**

Employee-Id: 1

Name: Rahul

Designation: Trainer

Department: IT

Salary: 60000

**Sample Input 2**

2

Shripad

HR

Admin

35000

**Sample Output 2**

Employee-Id: 2

Name: Shripad

Designation: HR

Department: Admin

Salary: 35000

**Input Explanation**

Input consists of 5 values

First input consists integer value that is Employee-Id

Second input consists string value that is Name

Third input consists String value that is Designation

Fourth input consists String value that is Department

Fifth input consists integer value that is Salary

**Output Explanation**

Output consists multiple integers & string value in specified form.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 3  Sanjay  Developer  IT  35000 | 4  Shekhar  Tester  IT  45000 | 5  Sangita  Senior-HR  Admin  25000 | 6  Kiran  Analyst  IT  35000 | 7  Jayashree  Accountant  Admin  35000 |
| **Output** | Employee-Id: 3  Name: Sanjay  Designation: Developer  Department: IT  Salary: 35000 | Employee-Id: 4  Name: Shekhar  Designation: Tester  Department: IT  Salary: 45000 | Employee-Id: 5  Name: Sangita  Designation: Senior-HR  Department: Admin  Salary: 25000 | Employee-Id: 6  Name: Kiran  Designation: Analyst  Department: IT  Salary: 35000 | Employee-Id: 7  Name: Jayashree  Designation: Accountant  Department: Admin  Salary: 35000 |

**#Solution**

#include<stdio.h>

struct employee

{

int id;

char name[20];

char designation[20];

char dept[20];

int sal;

};

void main()

{

struct employee emp;

scanf("%d",&emp.id);

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

scanf("%s",emp.designation);

scanf("%s",emp.dept);

scanf("%d",&emp.sal);

printf("Employee-Id: %d\n",emp.id);

printf("Name: %s\n",emp.name);

printf("Designation: %s\n",emp.designation);

printf("Department: %s\n",emp.dept);

printf("Salary: %d",emp.sal);

}

Q.2. Write a program in C to print first n natural numbers using recursion.

**Sample Input 1**

5

**Sample Output 1**

1 2 3 4 5

**Sample Input 2**

7

**Sample Output 2**

1 2 3 4 5 6 7

**Input Explanation**

Input consists of single integer value n

**Output Explanation**

Output consists of multiple sequential integer values upto n.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 6 | 8 | 5 | 4 | 3 |
| **Output** | 1 2 3 4 5 6 | 1 2 3 4 5 6 7 8 | 1 2 3 4 5 | 1 2 3 4 | 1 2 3 |

**#Solution**

#include<stdio.h>

int numPrint(int, int);

int main()

{

int n = 1;

int n1;

scanf("%d",&n1);

numPrint(n,n1);

return 0;

}

int numPrint(int n,int n1)

{

if(n<=n1)

{

printf("%d ",n);

numPrint(n+1,n1);

}

}

**1 Coding Question (10 mark)**

Q.1. Write a 'C' program to accept customer details such as: Account\_no, Name, Balance using structure. Assume 3 customers in the bank. Write a function to print the account no. and name of each customer whose balance < 1000 Rs, if no customer found whose balance is less than 1000, print None.

**Sample Input 1**

1

Hitesh

100000

2

Pravin

45632

3

Sanjay

500

**Sample Output 1**

3

Sanjay

500

**Sample Input 2**

1

Lalit

500

2

Rahil

900

3

Mahesh

800

**Sample Output 2**

1

Lalit

500

2

Rahil

900

3

Mahesh

800

**Input Explanation**

Input consists of three values for each customer

First input consists integer value which represents Account number of customer

Second input consists String value which represents Name of customer

Third input consists integer value which represents bank balance of customer

**Output Explanation**

Output consists of three values for each customer whose bank balance is less than 1000.

First output consists integer value which represents Account number of customer

Second output consists String value which represents Name of customer

Third output consists integer value which represents bank balance of customer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 1  Jiya  953  2  Khushi  620  3  Rahul  9000 | 1  Manoj  8600  2  Vinay  9530  3  Larisha  9006 | 1  Meena  9860  2  Ram  986  3  Charan  97862 | 13  Ravi  986  34  Manu  753  95  Anu  369 | 15  Raj  9879  17  Satish  6545  18  Mona  8532 |
| **Output** | 1  Jiya  953  2  Khushi  620 | None | 2  Ram  986 | 13  Ravi  986  34  Manu  753  95  Anu  369 | None |

**#Solution**

#include<stdio.h>

struct bank

{

int acc\_no;

char name[20];

int bal;

}b[3];

void check(struct bank b[],int n)

{

int i,c=0;

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

{

if(b[i].bal<1000)

{

printf("%d\n",b[i].acc\_no);

printf("%s\n",b[i].name);

printf("%d\n",b[i].bal);

c++;

}

}

if (c==0)

{

printf("None");

}

}

int main()

{

int i;

for(i=0;i<3;i++)

{

scanf("%d",&b[i].acc\_no);

scanf("%s",b[i].name);

scanf("%d",&b[i].bal);

}

check(b,3);

return 0;

}