

1. The following C declarations

```
struct node{  
    int i;  
    float j;  
};  
struct node *s[10];  
define s to be
```

- A. **An array, each element of which is a pointer to a structure of type node**
- B. A Structure of 2 fields, each field being a pointer to an array of 10 elements
- C. A Structure of 3 fields, an integer, a float, and an array of 10 elements
- D. An array, each element of which is a structure of type node

Answer: A

2. What will be the output of the following code?

```
struct student  
{  
    char *c;  
    struct student *point;  
};  
void main() {  
    struct student s;  
    printf("%d", sizeof(s));  
}
```

- A. 8
- B. 12
- C. 32
- D. 16**

Answer: D

3. What will be the output of the following code?

```
int main() {  
    struct forest  
    {  
        int trees;  
        int animals;  
    } F1,*F2;  
    F1.trees=1000;  
    F1.animals=20;  
    F2=&F1;  
    printf("%d ",F2.animals);  
}
```

- A. 1000
- B. 20
- C. Compile time error**
- D. 100

Answer: C

4. Can you combine the following two statements into one?

```
char *p;  
p = (char*) malloc(100);
```

- A. `char p = *malloc(100);`
- B. `char *p = (char) malloc(100);`
- C. `char *p = (char*)malloc(100);`**
- D. `char *p = (char *) (malloc*)(100);`

Answer: C

5. If a variable is a pointer to a structure, then which of the following operator is used to access data members of the structure through the pointer variable?

- A. `.`
- B. `&`
- C. `*`
- D. `->`**

Answer: D

6. What would be the equivalent pointer expression for referring the array element `a[i][j][k][l]`

- A. `((((a+i)+j)+k)+l)`
- B. `*(*(*(*a+i)+j)+k)+l`**
- C. `((((a+i)+j)+k)+l)`
- D. `((a+i)+j+k+l)`

Answer: B

7. What will be the output of the following code?

```
struct temp  
{  
    int a;  
    int b;  
    int c;  
};  
void main()  
{  
    struct temp p[] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};  
    printf("%d", p[2].a);  
}
```

- A. 3
- B. 6
- C. 7**
- D. 9

Answer: C

8. What will the outcome of the following code be?

```
struct student  
{  
    char *c;  
    struct student point;
```

```
};
void main()
{
    struct student s;
    s.c = "CSE";
    printf("%s", s.c);
}
```

- A. CSE
- B. Compile time error**
- C. Nothing
- D. 0

Answer: B

9. Choose a correct statement about C structure elements?
- a) Structure elements are stored on random free memory locations
  - b) structure elements are stored in register memory locations
  - c) structure elements are stored in contiguous memory locations
  - d) Structure elements can be initialized at the time of declaration.

Answer: C

10. What is the size of a C structure?
- a) C structure is always 128 bytes
  - b) Size of C structure is the total bytes of all elements of structure
  - c) Size of C structure is the size of largest elements
  - d) C structure is always 64 bytes

Answer: B

11. What is actually passed if you pass a structure variable to a function?
- a) Copy of structure variable
  - b) Reference of structure variable
  - c) Starting address of structure variable
  - d) Ending address of structure variable

Answer: A

12. The correct syntax to access the member of the ith structure in the array of structures is?

```
struct temp {
    int b;
}s[50];
```

- a) s.b[i];
- b) s[i].b;
- c) s.b[i];
- d) s[i].b;

Answer: D

13. Choose the correct output of the following code:

```
#include <stdio.h> struct student
{
    char *Student_name;
};
struct student s[2]; void main()
{
    s[0].Student_name = "FirstName"; s[1] = s[0];
```

```
printf("%s%s", s[0].Student_name, s[1].Student_name); s[1].Student_name =
"SecondName";
printf("%s%s", s[0].Student_name, s[1].Student_name);
}
```

- a) FirstName FirstName FirstName SecondName
- b) FirstName FirstName SecondName SecondName
- c) FirstName SecondName FirstName SecondName
- d) run time error

Answer: A

14. Which of the following operator connects the structure name to its member name?

- a) - (hyphen)
- b) <- (arrow)
- c) . (dot)
- d) Both <- and .

Answer: C

15. Choose the option which accesses the 8th student detail in the structure object "college\_student[10]".

- a. college\_student[7].details
- b. college\_student[9].details
- c. college\_student[8].details
- d. college\_students[8].details[8]

Answer: A

1. Define a structure called DSA with register number(integer), name of the student, cgpa. Write a C program to read the details of register number, name of the student, CGPA of 100 students and print the register number, name of the student and the cgpa whose cgpa is below 5.

/Function to find the details of student whose cgpa < 100./

void check(struct srm s[],int n) /Passing Array of structure to function/

```
{
    int i;
    for(i=0;i<n;i++)
    {
        if(s[i].cgpa<5)
        {
            printf("Account Number : %d\n",s[i].reg_no);
        }
    }
}
```

```

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

        printf("Balance   : %d\n",s[i].cgpa);

        printf("-----\n");

    }

}

}

int main()

{

    int i;

    for(i=0;i<100;i++)

    {

        printf("Enter Details of student %d\n",i+1);

        printf("-----\n");

        printf("Enter Register Number : ");

        scanf("%d",&s[i].reg_no);

        printf("Enter Name      : ");

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

        printf("Enter cgpa      : ");

        scanf("%d",&s[i].cgpa);

        printf("-----\n");

    }

    check(s,100);      //call function check

    return 0;

}

```

2. Anish have a fruits shop and for his customers he usually give manual bill. Kindly help him to generate computerized bill to his shop.

Input:

First line indicate name of the product

Second line indicates the number of items

Third line indicate price

Output:

First line indicate name of the product

Second line indicates the total amount in floating point with two dotted values.

Third indicate the 14% GST amount

Fourth indicate Total amount of the products

Answer:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
struct groceryshop
```

```
{
```

```
    int qty;
```

```
    char name[10];
```

```
    float price,gst;
```

```
};
```

```
int main()
```

```
{ struct groceryshop tax;
```

```
//char ch[10];
```

```
scanf("%s",tax.name);
```

```
scanf("%f %d",&tax.price,&tax.qty);
```

```
tax.gst=0.14;
```

```
float total=tax.price*tax.qty;
```

```
printf("%s\n%.2f\n%.2f\n%.2f",tax.name,total,total*tax.gst,total+(total*tax.gst));
```

```
return 0;
```

```
}
```

3. Lakshmi loves women's football. On the day of the women's football match in Tamil Nadu VS Manipur, She left home because of her friend's birthday and forgot to recharge his internet.

Due to this, she is very eager to go the home and watch the match, when she went home, Tamilnadu was playing.

So when she switched on the T,V, it is showing the points scored by an individual player in match, but she is unable to see the Total points scored by the team.

So, help her knowing the total points scored by the team by creating a program to accept shooting information of the football team using "Structure". It contains the player names and points scored by a player.

Constrain:

```
a[1]<=name<=a[100]
```

```
1<=runs<=100
```

Input Formats:

Get the string input represents "player name" and "points"

Output Formats:

Print the values "Total Points Scored".

Answer:

```

#include <stdio.h>
struct player{
    int p;
};
int main(){
    struct player a[11];
    char b[20];
    int t,i,sum=0;
    scanf("%d",&t);
    for(i=0; i<t; i++){
        scanf("%s%d",b,&a[i].p);
        sum+=a[i].p;
    }
    printf("Total Points:%d",sum);
    return 0;
}

```

~~4. Create a structure to specify data of customers in a bank. The data to be stored is: Account number, Name, Balance in account. Assume maximum of 20 customers in the bank. Create a function to read all customers details and call it in main. Your program must be menu driven with following options~~

4. Create a structure to specify data of customers in a bank. The data to be stored is: Account number, Name, Balance in account. Assume maximum of 20 customers in the bank. Create a function to read all customers details and call it in main. Your program must be menu driven with following options

1. Print the Account number and name and balance of each customer.
2. Withdraw money
3. Deposit money
4. Search Customer

```

struct customer

```

```

{

```

```

    int account_no;

```

```

    char name[80];

```

```

    int balance;

```

```

};

```

```

void accept(struct customer[], int);

```

```

void display(struct customer[], int);

```

```

int search(struct customer[], int, int);

```

```

void deposit(struct customer[], int, int, int);

```

```

void withdraw(struct customer[], int, int, int);

```

```

int main()

```

```
{  
  
    struct customer data[20];  
  
    int n, choice, account_no, amount, index;  
  
  
    printf("Banking System\n\n");  
  
    printf("Number of customer records you want to enter? : ");  
  
    scanf("%d", &n);  
  
    accept(data, n);  
  
    do  
    {  
  
        printf("\nBanking System Menu :\n");  
  
        printf("Press 1 to display all records.\n");  
  
        printf("Press 2 to search a record.\n");  
  
        printf("Press 3 to deposit amount.\n");  
  
        printf("Press 4 to withdraw amount.\n");  
  
        printf("Press 0 to exit\n");  
  
        printf("\nEnter choice(0-4) : ");  
  
        scanf("%d", &choice);  
  
        switch (choice)  
        {  
  
            case 1:  
  
                display(data, n);  
  
                break;  
  
            case 2:  
  
                printf("Enter account number to search : ");
```



```
scanf("%d", &account_no);

index = search(data, n, account_no);

if (index == - 1)

{

    printf("Record not found : ");

}

else

{

    printf("A/c Number: %d\nName: %s\nBalance: %d\n",

        data[index].account_no, data[index].name,

        data[index].balance);

}

break;

case 3:

    printf("Enter account number : ");

    scanf("%d", &account_no);

    printf("Enter amount to deposit : ");

    scanf("%d", &amount);

    deposit(data, n, account_no, amount);

    break;

case 4:

    printf("Enter account number : ");

    scanf("%d", &account_no);

    printf("Enter amount to withdraw : ");

    scanf("%d", &amount);
```

```

        withdraw(data, n, account_no, amount);

    }

}

while (choice != 0);

return 0;

}

void accept(struct customer list[80], int s)

{

    int i;

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

    {

        printf("\nEnter data for Record #%d", i + 1);

        printf("\nEnter account_no : ");

        scanf("%d", &list[i].account_no);

        fflush(stdin);

        printf("Enter name : ");

        gets(list[i].name);

        list[i].balance = 0;

    }

}

void display(struct customer list[80], int s)

{

    int i;

    printf("\n\nA/c No\tName\tBalance\n");

```

```
    for (i = 0; i < s; i++)  
    {  
        printf("%d\t%s\t%d\n", list[i].account_no, list[i].name,  
            list[i].balance);  
    }  
}  
  
int search(struct customer list[80], int s, int number)  
{  
    int i;  
    for (i = 0; i < s; i++)  
    {  
        if (list[i].account_no == number)  
        {  
            return i;  
        }  
    }  
    return - 1;  
}  
  
void deposit(struct customer list[], int s, int number, int amt)  
{  
    int i = search(list, s, number);  
    if (i == - 1)  
    {  
        printf("Record not found");  
    }  
}
```

```

        else

        {

            list[i].balance += amt;

        }

    }

void withdraw(struct customer list[], int s, int number, int amt)

{

    int i = search(list, s, number);

    if (i == - 1)

    {

        printf("Record not found\n");

    }

    else if (list[i].balance < amt)

    {

        printf("Insufficient balance\n");

    }

    else

    {

        list[i].balance -= amt;

    }

}

```

5. Write a C program to read and print employee details using structure.

To store multiple employee details we will use an array of structures. Each element in the array will represent a single employee.

Each Structure i.e. *Employee* contains:

- Name
- Id

- Salary

```
typedef struct{
    char name[30];
    int id;
    double salary;
} Employee;

int main()
{
    //number of employees
    int n=2;

    //array to store structure values of all employees
    Employee employees[n];

    //Taking each employee detail as input
    printf("Enter %d Employee Details \n \n",n);
    for(int i=0; i<n; i++){
        printf("Employee %d:- \n",i+1);

        //Name
        printf("Name: ");
        scanf("%[^\\n]s",employees[i].name);

        //ID
        printf("Id: ");
        scanf("%d",&employees[i].id);

        //Salary
        printf("Salary: ");
        scanf("%lf",&employees[i].salary);

        //to consume extra '\\n' input
        char ch = getchar();

        printf("\\n");
    }

    //Displaying Employee details
    printf("----- All Employees Details -----\\n");
    for(int i=0; i<n; i++){
        printf("Name \\t: ");
        printf("%s \\n",employees[i].name);

        printf("Id \\t: ");
        printf("%d \\n",employees[i].id);

        printf("Salary \\t: ");
        printf("%.2lf \\n",employees[i].salary);
    }
}
```

```

printf("\n");
}

return 0;
}

```

And to sort the players by runs scored, we can use the following code:

6. Write a C program to store the player name , runs scored , wickets taken of a cricket player using structure and sort the players in order of minimum runs to maximum runs

```

struct cricketer
{
    int runs,wickets;
    char name[25];
}player[100],t;

void main()
{
    int i,j,n;

    printf("Enter the no of cricket players\n");

    scanf("%d",&n);

    printf("Enter player info as name , wickets taken , runs scored\n");

    for(i=0;i<n;i++)
    {
        scanf("%s %d %d",player[i].name,&player[i].wickets,&player[i].runs);
    }

    for(i=0;i<n;i++)
    {
        for(j=0;j<n-1;j++)
        {
            if(player[j].runs>player[j+1].runs)
            {

```

```
        t=player[j];
        player[j]=player[j+1];
        player[j+1]=t;
    }
}
}

printf("\nCricketer info in terms of runs scored from lowest to highest\n");
printf("\nNAME\t\tWICKETS\t\tRUNS_SCORED\n");
printf("-----\n");
for(i=0;i<n;i++)
{
    printf("%s\t\t%d\t\t%d\n",player[i].name,player[i].wickets,player[i].runs);
}
}
```