ACP ASSIGNMENT

20BCA053

Jaison John

Q. Write a program to store Information (name, roll and marks) of a Student Using Structure

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

struct info
{
    char name[25];
    int roll_no;
    float marks;
};

void main()
{
    struct info i;

    printf("\nEnter Name:\t");
    scanf("%c",i.name);
    fflush(stdin);
    printf("\nEnter Roll no:\t");
    scanf("%d",&i.roll_no);
    printf("\nEnter Marks:\t");
    scanf("%f",&i.marks);
    getch();
}
```

```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\jaiso\Desktop\ACP> cd "c:\Users\jaiso\Desktop\ACP\" ; if ($?) { gcc 1.c -o 1 } ; if ($?) {

Enter Name:     Jaison John

Enter Roll no:  53

Enter Marks:    97
```

Q. Define a structure called distance. Add Two Distances (in inch-feet) System Using Structures

```c
#include <stdio.h>
struct Distance {
    int feet;
    float inch;
} d1, d2, result;
int main()
{
    printf("Enter 1st distance\n");
    printf("Enter feet: ");
    scanf("%d", &d1.feet);
    printf("Enter inch: ");
    scanf("%f", &d1.inch);

    printf("\nEnter 2nd distance\n");
    printf("Enter feet: ");
    scanf("%d", &d2.feet);
    printf("Enter inch: ");
    scanf("%f", &d2.inch);

result.feet = d1.feet + d2.feet;
result.inch = d1.inch + d2.inch;
result.inch = result.inch - 12.0;
++result.feet;
printf("\nSum of distances = %d\'-%.1f\"", result.feet, result.inch);
return 0;
}
```

```
PS C:\Users\jaiso\Desktop\ACP> cd "c:\Users\jaiso\Desktop\ACP\" ; if ($?) { gcc 2.c -o 2 } ; if ($?)
Enter 1st distance
Enter feet: 5
Enter inch: 12

Enter 2nd distance
Enter feet: 6
Enter inch: 14

Sum of distances = 12'-14.0"
PS C:\Users\jaiso\Desktop\ACP>
Ln 30, Col 1 (633 selected)  Spa
```

Q. Write a program to add Two Complex Numbers by Passing Structure to a Function

```c
#include <stdio.h>
typedef struct complex {
float real;
float imag;
} complex;
complex add(complex n1, complex n2);
int main()
{
complex n1, n2, result;
printf("For 1st complex number \n");
printf("Enter the real and imaginary parts: ");
scanf("%f %f", &n1.real, &n1.imag);

printf("\nFor 2nd complex number \n");
printf("Enter the real and imaginary parts: ");
scanf("%f %f", &n2.real, &n2.imag);
result = add(n1, n2);
printf("Sum = %.1f + %.1fi", result.real, result.imag);
return 0;
}
complex add(complex n1, complex n2)
{
complex temp;
temp.real = n1.real + n2.real;
temp.imag = n1.imag + n2.imag;
return (temp);
}
```

```
PS C:\Users\jaiso\Desktop\ACP> cd "c:\Users\jaiso\Desktop\ACP\" ; if (
For 1st complex number
Enter the real and imaginary parts: 5.6
5.2

For 2nd complex number
Enter the real and imaginary parts: 2.2
2.5
Sum = 7.8 + 7.7i
PS C:\Users\jaiso\Desktop\ACP>
```

Q. Calculate Difference Between Two Time Period

```c
#include <stdio.h>
struct TIME
{
    int seconds;
    int minutes;
    int hours;
};

void differenceBetweenTimePeriod(struct TIME t1,
                                 struct TIME t2, struct TIME *diff);

int main()
{
    struct TIME startTime, stopTime, diff;

    printf("Enter the start time. \n");
    printf("Enter hours, minutes and seconds: ");
    scanf("%d %d %d", &startTime.hours, &startTime.minutes,
          &startTime.seconds);

    printf("Enter the stop time. \n");
    printf("Enter hours, minutes and seconds: ");
    scanf("%d %d %d", &stopTime.hours,
          &stopTime.minutes,
          &stopTime.seconds);

    differenceBetweenTimePeriod(startTime, stopTime, &diff);
    printf("\nTime Difference: %d:%d:%d - ", startTime.hours,
startTime.minutes,
startTime.seconds);
```

```
printf("%d:%d:%d ", stopTime.hours,
       stopTime.minutes, stopTime.seconds);
printf("=\t %d:%d:%d\n", diff.hours, diff.minutes,
       diff.seconds);
return 0;
}

void differenceBetweenTimePeriod(struct TIME start,
                                 struct TIME stop, struct TIME *diff)
{
    while (stop.seconds > start.seconds)
    {
        --start.minutes;
        start.seconds += 60;
    }
    diff->seconds = start.seconds - stop.seconds;
    while (stop.minutes > start.minutes)
    {
        --start.hours;
        start.minutes += 60;
    }
    diff->minutes = start.minutes - stop.minutes;
    diff->hours = start.hours - stop.hours;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\jaiso\Desktop\ACP> cd "c:\Users\jaiso\Desktop\ACP\" ; if ($?)
Enter the start time.
Enter hours, minutes and seconds: 1 20 100
Enter the stop time.
Enter hours, minutes and seconds: 1 05 100

Time Difference: 1:20:100 - 1:5:100 =    0:15:0
PS C:\Users\jaiso\Desktop\ACP>
```

Q. Store Information of 10 Students Using Structure

```c
#include <stdio.h>
struct student
{
    char firstName[30];
    int roll;
    float marks;
} s[10];

int main()
{
    printf("Enter information of students:\n");
    for (int i = 0; i < 10; ++i)
    {
        s[i].roll = i + 1;
        printf("\nFor roll number%d,\n", s[i].roll);
        printf("Enter first name: ");
        scanf("%s", s[i].firstName);
        fflush(stdin);
        printf("Enter marks: ");
        scanf("%f", &s[i].marks);
    }
    printf("\t\tDisplaying Information:\n\n");

    for (int i = 0; i < 10; ++i)
    {
        printf("\nRoll number: %d\t\t", i + 1);
        printf("First name: ");
        puts(s[i].firstName);
        printf("\tMarks: %.2f\n", s[i].marks);
    }}
```

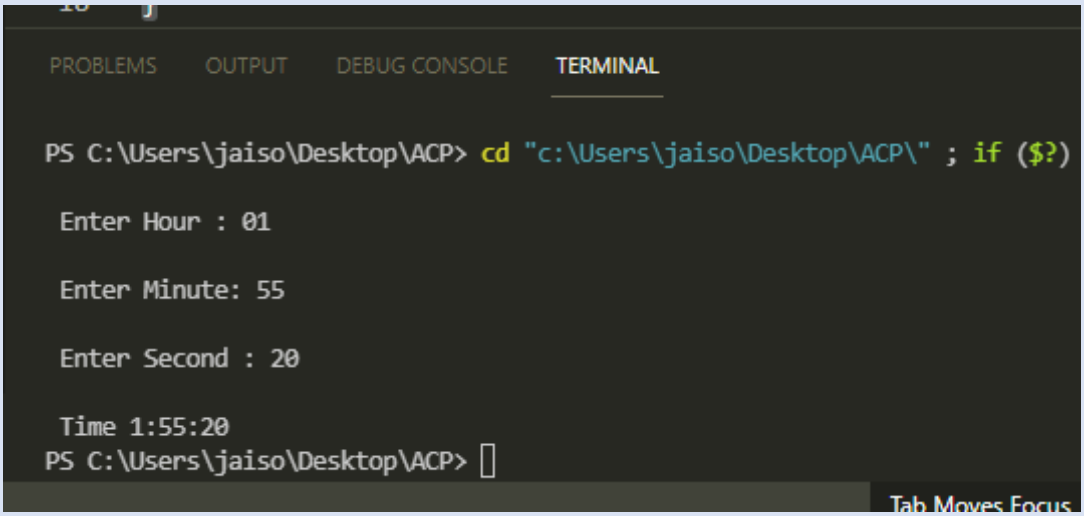
```
                Displaying Information:


Roll number: 1          First name: Jaison
        Marks: 98.80

Roll number: 2          First name: Prem
        Marks: 98.80

Roll number: 3          First name: Ankush
        Marks: 95.90

Roll number: 4          First name: Able
        Marks: 89.20

Roll number: 5          First name: Devansh
        Marks: 98.70

Roll number: 6          First name: Manav
        Marks: 50.10

Roll number: 7          First name: Rohit
        Marks: 70.40

Roll number: 8          First name: Seteven
        Marks: 60.80

Roll number: 9          First name: Rohit
        Marks: 54.20

Roll number: 10         First name: Mickey
        Marks: 40.00
PS C:\Users\jaiso\Desktop\ACP>
```

Q.

Define a structure data type called time struct containing three members : hour, minute, and second. Create a function to input values to members and another function to display values.

```c
#include <stdio.h>
struct time_struct
{
    int hour;
    int minute;
    int second;
}t;
int main()
{
    printf("\n Enter Hour : ");
    scanf("%d", &t.hour);
    printf("\n Enter Minute: ");
    scanf("%d", &t.minute);
```

```
    printf("\n Enter Second : ");
    scanf("%d", &t.second);
    printf("\n Time %d:%d:%d", t.hour % 24, t.minute % 60, t.second % 60);
    return 0;
}
```



Q. Define a structure named census with the following memebers: city, population, and literacy level. Write a function to store the list alphabetically.

```
#include <stdio.h>
struct census
{
    char city[50];
    long int population;
    float li;
};

void main()
{
```

```
    struct census temp, c[5];
    for (i = 0; i < 5; i++)
    {
        printf("\nenter city name, population and literacy level");
        scanf("%s%ld%f", &c[i].city, &c[i].population, &c[i].li);
    }
    for (int p = 0; p < 4; p++)
    {
        for (int j = 0; j < 5 - p - 1; j++)
        {
            if (c[j].li > c[j + 1].li)
            {
                temp = c[j];
                c[j] = c[j + 1];
                c[j + 1] = temp;
            }
        }
    }
    printf("\nsorted order are\n");
    for (int i = 0; i < 5; i++)
    {
        printf("%s\t\t\t\t%ld\t\t%.2f%%\n", c[i].city, c[i].population, c[i].li);
    }
}
```

```
enter city name, population and literacy level
Kerela
6849651
100

enter city name, population and literacy level
UP
68461321
70

enter city name, population and literacy level
Dehli
986465
80

enter city name, population and literacy level
Goa
4656
90

sorted order are
UP                              68461321                70.00%
Mumbai                          986465          80.00%
Dehli                           986465          80.00%
Goa                             4656            90.00%
Kerela                          6849651         100.00%
PS C:\Users\jaiso\Desktop\ACP>
```

Q. Write a program in C to show the basic declaration of pointer

```c
#include <stdio.h>
int main()
{
    int x=20;
    int *ptr;

    ptr=&x;

    printf("Memory address of x: %p\n",ptr);
    printf("Value x: %d\n",*ptr);
    return 0;
}
```

```
Memory address of x: 0061FEC8
Value x: 20
```

Q. Write a program in C to demonstrate how to handle the pointers in the program

```c
#include <stdio.h>
int main()
{
    int *ab;
    int m;
    m = 29;
    printf(" Here in the declaration ab = int pointer, int m= 29\n\n");
    printf(" Address of m : %p\n", &m);
    printf(" Value of m : %d\n\n", m);
    ab = &m;
    printf(" Now ab is assigned with the address of m.\n");
    printf(" Address of pointer ab : %p\n", ab);
    printf(" Content of pointer ab : %d\n\n", *ab);
    m = 34;
    printf(" The value of m assigned to 34 now.\n");
    printf(" Address of pointer ab : %p\n", ab);
    printf(" Content of pointer ab : %d\n\n", *ab);
    *ab = 7;
    printf(" The pointer variable ab is assigned the value 7 now.\n");
    printf(" Address of m : %p\n", &m);
    printf(" Value of m : %d\n\n", m);
    return 0;
}
```

```
Here in the declaration ab = int pointer, int m= 29

Address of m : 0061FEC8
Value of m : 29

Now ab is assigned with the address of m.
Address of pointer ab : 0061FEC8
Content of pointer ab : 29

The value of m assigned to 34 now.
Address of pointer ab : 0061FEC8
Content of pointer ab : 34

The pointer variable ab is assigned the value 7 now.
Address of m : 0061FEC8
Value of m : 7

PS C:\Users\jaiso\Desktop\ACP>
```

Q. Write a program in C to demonstrate the use of &(address of) and *(value at address) operator

```c
#include <stdio.h>
void main()
{
    int m = 300;
    float fx = 300.60;
    char cht = 'J';
    int *pt1;
    float *pt2;
    char *pt3;
    pt1 = &m;
    pt2 = &fx;
    pt3 = &cht;
    printf(" m = %d\n", m);
    printf(" fx = %f\n", fx);
    printf(" cht = %c\n", cht);
    printf("\n Using & operator :\n");
    printf("-----------------------\n");
```

```
    printf(" address of m = %p\n", &m);
    printf(" address of fx = %p\n", &fx);
    printf(" address of cht = %p\n", &cht);
    printf("\n Using & and * operator :\n");
    printf("----------------------------\n");
    printf(" value at address of m = %d\n", *(&m));

    printf(" value at address of fx = %f\n", *(&fx));
    printf(" value at address of cht = %c\n", *(&cht));
    printf("\n Using only pointer variable :\n");
    printf("---------------------------------\n");
    printf(" address of m = %p\n", pt1);
    printf(" address of fx = %p\n", pt2);
    printf(" address of cht = %p\n", pt3);
    printf("\n Using only pointer operator :\n");
    printf("---------------------------------\n");
    printf(" value at address of m = %d\n", *pt1);
    printf(" value at address of fx= %f\n", *pt2);
    printf(" value at address of cht= %c\n\n", *pt3);
}
```

```
 m = 300
 fx = 300.600006
 cht = J

 Using & operator :
----------------------
 address of m = 0061FEC0
 address of fx = 0061FEBC
 address of cht = 0061FEBB

 Using & and * operator :
---------------------------
 value at address of m = 300
 value at address of fx = 300.600006
 value at address of cht = J

 Using only pointer variable :
--------------------------------
 address of m = 0061FEC0
 address of fx = 0061FEBC
 address of cht = 0061FEBB

 Using only pointer operator :
--------------------------------
 value at address of m = 300
 value at address of fx= 300.600006
 value at address of cht= J
```

Q. Write a program in C to add two numbers using pointers.

```c
#include <stdio.h>
int main()
{
    int fno, sno, *ptr, *qtr, sum;
    printf(" Input the first number : ");
    scanf("%d", &fno);
    printf(" Input the second number : ");
    scanf("%d", &sno);

    ptr = &fno;
    qtr = &sno;

    sum = *ptr + *qtr;

    printf(" The sum of the entered numbers is : %d\n\n", sum);

    return 0;
}
```

```
PS C:\Users\jaiso\Desktop\ACP> cd "c:\Users\jaiso\D
 Input the first number : 50
 Input the second number : 80
 The sum of the entered numbers is : 130

PS C:\Users\jaiso\Desktop\ACP>
```

Q. Write a program in C to find the maximum number between two numbers using a pointer

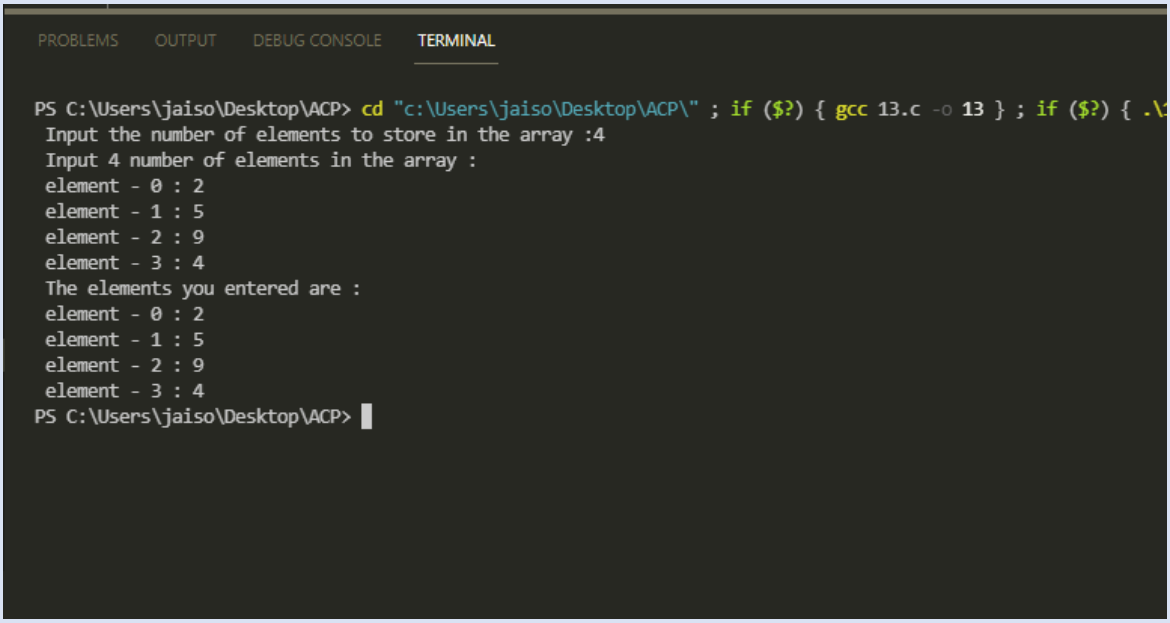
```c
#include <stdio.h>
#include <stdlib.h>
void main()
{
    int fno, sno, *ptr1 = &fno, *ptr2 = &sno;
    printf(" Input the first number : ");
    scanf("%d", ptr1);
    printf(" Input the second number : ");
    scanf("%d", ptr2);

    if (*ptr1 > *ptr2)
        printf("\n\n %d is the maximum number.\n\n", *ptr1);

    else
        printf("\n\n %d is the maximum number.\n\n", *ptr2);
}
```

```
 Input the first number : 50
 Input the second number : 20


 50 is the maximum number.
```

Q. Write a program in C to store n elements in an array and print the elements using pointer.

```c
#include <stdio.h>
int main()
{
    int arr1[25], n;
    printf(" Input the number of elements to store in the array :");
    scanf("%d", &n);
    printf(" Input %d number of elements in the array :\n", n);
    for (int i = 0; i < n; i++)
    {
        printf(" element - %d : ", i);
        scanf("%d", arr1 + i);
```

```
    }
    printf(" The elements you entered are : \n");
    for (int i = 0; i < n; i++)
    {
        printf(" element - %d : %d \n", i, *(arr1 + i));
    }
    return 0;
}
```



Q. Write a program in C to find the largest element using Dynamic Memory Allocation.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int i, n;
    float *element;
    printf(" Input total number of elements(1 to 100): ");
    scanf("%d", &n);
```

```
    element = (float *)calloc(n, sizeof(float));
    if (element == NULL)
    {
        printf(" No memory is allocated.");
        exit(0);
    }
    printf("\n");
    for (i = 0; i < n; ++i)
    {
        printf(" Number %d: ", i + 1);
        scanf("%f", element + i);
    }
    for (i = 1; i < n; ++i)
    {

        if (*element < *(element + i))
            *element = *(element + i);
    }
    printf(" The Largest element is : %.2f \n\n", *element);
    return 0;
}
```

```
Input total number of elements(1 to 100): 4

Number 1: 50
Number 2: 22
Number 3: 19
Number 4: 90
The Largest element is : 90.00
```

Q. Write a program in C to Calculate the length of the string using a pointer

```
#include <stdio.h>
int calculateLength(char *);

void main()
{
```

```
    char str1[25];
    int l;
    printf(" Input a string : ");
    fgets(str1, sizeof str1, stdin);
    l = calculateLength(str1);
    printf(" The length of the given string %s is : %d ", str1, l - 1);
    printf("\n\n");
}
int calculateLength(char* ch)
{
    int ctr = 0;
    while (*ch != '\0')
    {
        ctr++;
        ch++;
    }
    return ctr;
}
```

```
PS C:\Users\jaiso\Desktop\ACP> cd "c:\Users\jaiso\Desktop\ACP\" ; if ($?) { gcc 15.c -o 15 } ; if ($?) { .\15 }
 Input a string : Jaison JOhn
 The length of the given string Jaison JOhn
 is : 11

PS C:\Users\jaiso\Desktop\ACP>
```

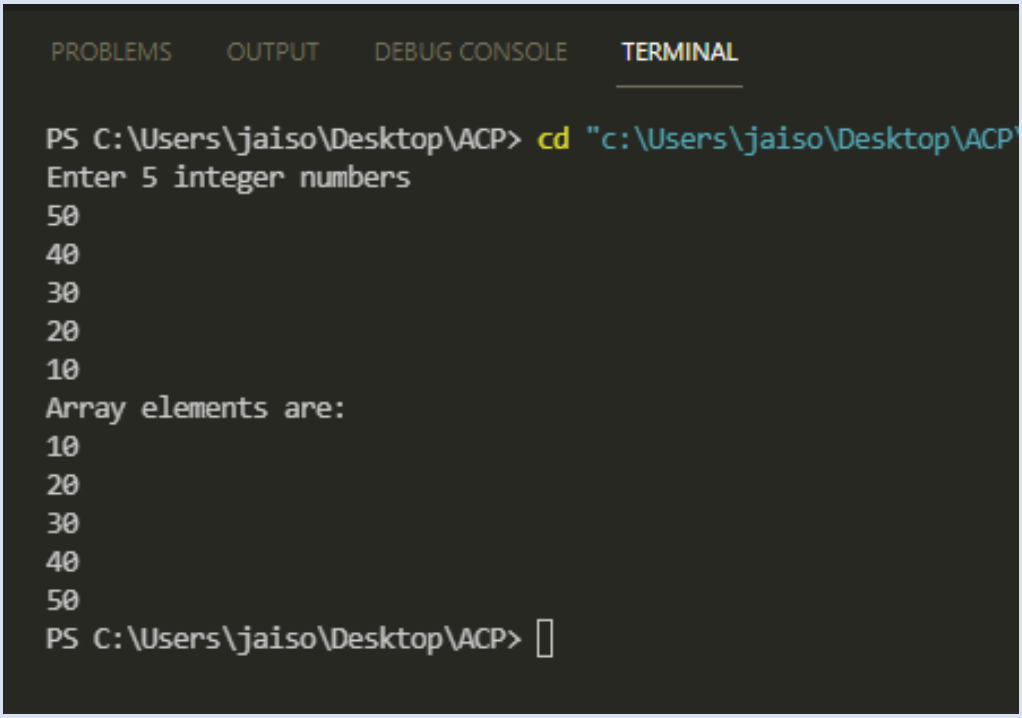
Q. Write a program in C to find the factorial of a given number using pointers

```c
#include <stdio.h>
void findFact(int,int*); int main()
{
    int fact;
    int num1;
    printf(" Input a number : ");
    scanf("%d", &num1);
    findFact(num1, &fact);
    printf(" The Factorial of %d is : %d \n\n", num1, fact);
    return 0;
}

void findFact(int n, int *f)
{
    int i;

    *f = 1;
    for (i = 1; i <= n; i++)
        *f = *f * i;
}
```

```
PS C:\Users\jaiso\Desktop\ACP> cd "c:\Users\jai
 Input a number : 5
 The Factorial of 5 is : 120

PS C:\Users\jaiso\Desktop\ACP>
```

Q. Write a program in C to print the elements of an array in reverse order

```c
#include <stdio.h>
int main()
{
    int a[5], i;
    printf("Enter 5 integer numbers\n");
    for (i = 0; i < 5; i++)
        scanf("%d", &a[i]);

    printf("Array elements are:\n");
    for (i = 4; i >= 0; i--)
        printf("%d\n", a[i]);
    return 0;
}
```



Q. Write a program to create and display singly link list using pointers

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

struct node
{
    int num;
    struct node *nextptr;
}*stnode;

void createNodeList(int n);
void displayList();

int main()
{
    int n;
        printf("\n\n Linked List - To create and display Singly Linked List :\n")
;
    printf(" Input the number of nodes : ");
    scanf("%d", &n);
    createNodeList(n);
    printf("\n Data entered in the list : \n");
    displayList();
    return 0;
}
void createNodeList(int n)
{
    struct node *fnNode, *tmp;
    int num, i;
    stnode = (struct node *)malloc(sizeof(struct node));

    if(stnode == NULL)
    {
        printf(" Memory can not be allocated.");
    }
    else
    {
        printf(" Input data for node 1 : ");
        scanf("%d", &num);
        stnode->num = num;
        stnode->nextptr = NULL;
        tmp = stnode;
```

```
        for(i=2; i<=n; i++)
        {
            fnNode = (struct node *)malloc(sizeof(struct node));
            if(fnNode == NULL)
            {
                printf(" Memory can not be allocated.");
                break;
            }
            else
            {
                printf(" Input data for node %d : ", i);
                scanf(" %d", &num);

                fnNode->num = num;
                fnNode->nextptr = NULL;

                tmp->nextptr = fnNode;
                tmp = tmp->nextptr;
            }
        }
    }
}
void displayList()
{
    struct node *tmp;
    if(stnode == NULL)
    {
        printf(" List is empty.");
    }
    else
    {
        tmp = stnode;
        while(tmp != NULL)
        {
            printf(" Data = %d\n", tmp->num);
            tmp = tmp->nextptr;
        }
    }
}
```

❖ Write a program to insert a node at the beginning of a singly linked list

```
Linked List - To create and display Singly Linked List :
Input the number of nodes : 4
Input data for node 1 : 50
Input data for node 2 : 80
Input data for node 3 : 40
Input data for node 4 : 90

Data entered in the list :
Data = 50
Data = 80
Data = 40
Data = 90
```

- **Write a program to insert a node at the end of a singly linked list**
- **Write a program to delete a node from the beginning of a singly linked list**
- **Write a program to from a node of the end of a singly linked list**

```c
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int data;
    struct node *next;
};
struct node *head;

void beginsert();
void lastinsert();

void begin_delete();
void last_delete();

void display();
void search();
void main()
{
    int choice = 0;
    while (choice != 9)
    {
        printf("\nChoose one option from the following list ...\n");
        printf("\n===============================================\n");
        printf("\n1.Insert in begining\n2.Insert at last\n3.Delete from Beginning
\n  4.Delete from last\n5.Search for an element\n6.Show\n7.Exit\n");
        printf("\nEnter your choice?\n");
```

```
        scanf("\n%d", &choice);
        switch (choice)
        {
        case 1:
            beginsert();
            break;
        case 2:
            lastinsert();
            break;
        case 3:
            begin_delete();
            break;
        case 4:
            last_delete();
            break;
        case 5:
            search();
            break;
        case 6:
            display();
            break;
        case 7:
            exit(0);
            break;
        default:
            printf("Please enter valid choice..");
        }
    }
}
void beginsert()
{
    struct node *ptr;
    int item;
    ptr = (struct node *)malloc(sizeof(struct node *));
    if (ptr == NULL)
    {
        printf("\nOVERFLOW");
    }
    else
    {
        printf("\nEnter value\n");
        scanf("%d", &item);
        ptr->data = item;
        ptr->next = head;
        head = ptr;
        printf("\nNode inserted");
    }
```

```
}
void lastinsert()
{
    struct node *ptr, *temp;
    int item;
    ptr = (struct node *)malloc(sizeof(struct node));
    if (ptr == NULL)
    {
        printf("\nOVERFLOW");
    }
    else
    {
        printf("\nEnter value?\n");
        scanf("%d", &item);
        ptr->data = item;
        if (head == NULL)
        {
            ptr->next = NULL;
            head = ptr;
            printf("\nNode inserted");
        }
        else
        {
            temp = head;
            while (temp->next != NULL)
            {
                temp = temp->next;
            }
            temp->next = ptr;
            ptr->next = NULL;
            printf("\nNode inserted");
        }
    }
}
void begin_delete()
{
    struct node *ptr;
    if (head == NULL)
    {
        printf("\nList is empty\n");
    }
    else
    {
        ptr = head;
        head = ptr->next;
        free(ptr);
        printf("\nNode deleted from the begining ...\n");
```

```
    }
}
void last_delete()
{
    struct node *ptr, *ptr1;
    if (head == NULL)
    {
        printf("\nlist is empty");
    }
    else if (head->next == NULL)
    {
        head = NULL;
        free(head);
        printf("\nOnly node of the list deleted ...\n");
    }

    else
    {
        ptr = head;
        while (ptr->next != NULL)
        {
            ptr1 = ptr;
            ptr = ptr->next;
        }
        ptr1->next = NULL;
        free(ptr);
        printf("\nDeleted Node from the last ...\n");
    }
}
void search()
{
    struct node *ptr;
    int item, i = 0, flag;
    ptr = head;
    if (ptr == NULL)
    {
        printf("\nEmpty List\n");
    }
    else
    {
        printf("\nEnter item which you want to search?\n");
        scanf("%d", &item);
        while (ptr != NULL)
        {
            if (ptr->data == item)
            {
                printf("item found at location %d ", i + 1);
```

```
                flag = 0;
            }
            else
            {
                flag = 1;
            }
            i++;
            ptr = ptr->next;
        }
        if (flag == 1)
        {
            printf("Item not found\n");
        }
    }
}

void display()
{
    struct node *ptr;
    ptr = head;
    if (ptr == NULL)
    {
        printf("Nothing to print");
    }
    else
    {
        printf("\nprinting values . . . . .\n");
        while (ptr != NULL)
        {
            printf("\n%d", ptr->data);
            ptr = ptr->next;
        }
    }
}
```

```
Choose one option from the following list ...

==============================================

1.Insert in begining
2.Insert at last
3.Delete from Beginning
  4.Delete from last
5.Search for an element
6.Show
7.Exit

Enter your choice?
6

printing values . . . . .

70
25
80
90
50
50
Choose one option from the following list ...

==============================================

1.Insert in begining
2.Insert at last
3.Delete from Beginning
  4.Delete from last
```

Ln 48, Col 15