Name: Aritra Das

Roll- 75

ASSIGNMENT 1:

1.

#include <stdio.h>

#include <conio.h>

void main()

{

    int a=15;

    int \*p;

    int \*\*pp;

    p=&a;

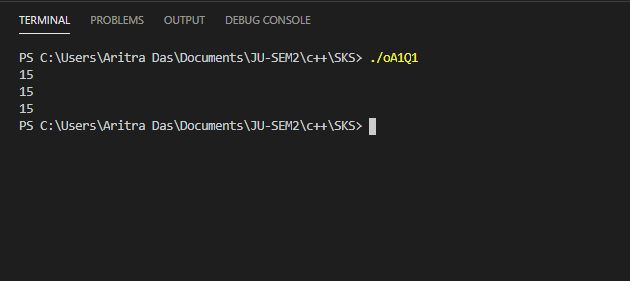
    pp=&p;

    printf("%d\n",a);

    printf("%d\n",\*p);

    printf("%d\n",\*\*pp);

}



2.

#include <stdio.h>

#include <conio.h>

void main()

{

    int\* ptr;

    int n, i;

    printf("Enter number of elements:\n");

    scanf("%d",&n);

    ptr = (int\*)malloc(n \* sizeof(int));

    if (ptr == NULL) {

        printf("Memory not allocated\n");

        exit(0);

    }

    else {

        printf("Memory successfully allocated using malloc.\n");

        for (i = 0; i < n; ++i) {

            printf("Enter element %d",i+1);

            scanf("%d",(ptr+i));

        }

        printf("The elements of the array are: ");

        for (i = 0; i < n; ++i) {

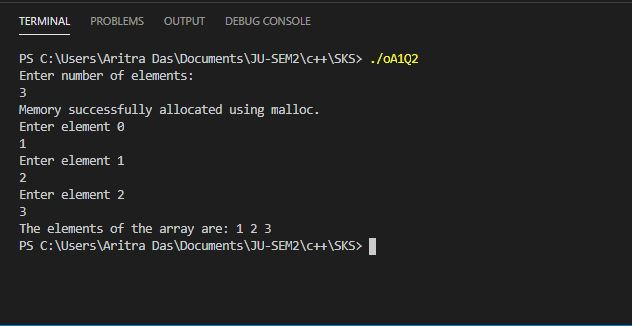
            printf("%d" ,\*(ptr+i));

        }

    }

    return 0;

}



3.

#include <stdio.h>

#include <conio.h>

void aop(){

    int r, c, i, j, count;

    printf("Enter rows and columns\n");

    scanf("%d%d",&r,&c);

    int \*arr[r];

    for (i=0; i<r; i++)

         arr[i] = (int \*)malloc(c \* sizeof(int));

    printf("Enter the elements");

    count = 0;

    for (i = 0; i <  r; i++)

      for (j = 0; j < c; j++)

         scanf("%d",&arr[i][j]);

    for (i = 0; i <  r; i++)

      for (j = 0; j < c; j++)

         printf("%d",arr[i][j]);

         printf("\n");

}

void sp(){

    int r, c, i, j, count=0;

    printf("Enter rows and columns\n");

    scanf("%d%d",&r,&c);

    int \*arr = (int \*)malloc(r \* c \* sizeof(int));

    printf("Enter the elements");

    for (i = 0; i <  r; i++)

      for (j = 0; j < c; j++)

         scanf("%d",(arr + i\*c + j));

    for (i = 0; i <  r; i++)

      for (j = 0; j < c; j++)

         printf("%d",\*(arr + i\*c + j));

         printf("\n");

}

int main()

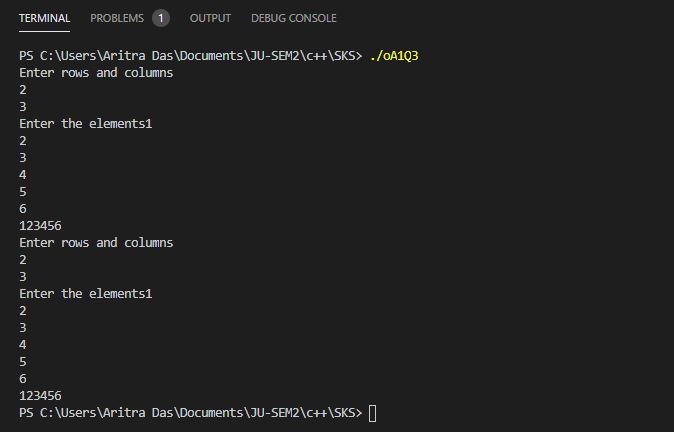
{

    aop();

    sp();

    return 0;

}



4.

#include <stdio.h>

#include <stdlib.h>

void read\_dim(int \*, int \*);

void dm\_alloc(int \*\*, int);

void read\_val(int \*\*, int, int);

void print\_val(int \*\*, int, int);

int main()

{

    int r, c, \*b[30];

    read\_dim(&r, &c);

    for (int i = 0; i < r; ++i){

        dm\_alloc(b + i, c);

    }

    read\_val(b, r, c);

    print\_val(b, r, c);

    for (int i = 0; i < r; ++i)

        free(b[i]);

    return 0;

}

void read\_dim(int \*m, int \*n)

{

    printf("Enter the size of row and column  of array\n");

    scanf("%d%d", m, n);

}

void dm\_alloc(int \*\*ele, int len)

{

    \*ele = (int \*)malloc(len \* sizeof(int));

}

void read\_val(int \*\*arr, int m, int n)

{

    printf("Enter values of array elements \n");

    for (int i = 0; i < m; ++i)

        for (int j = 0; j < n; ++j)

            scanf("%d", arr[i] + j);

}

void print\_val(int \*\*arr, int m, int n)

{

    printf("Values of the elements of the array\n");

    for (int i = 0; i < m; ++i)

    {

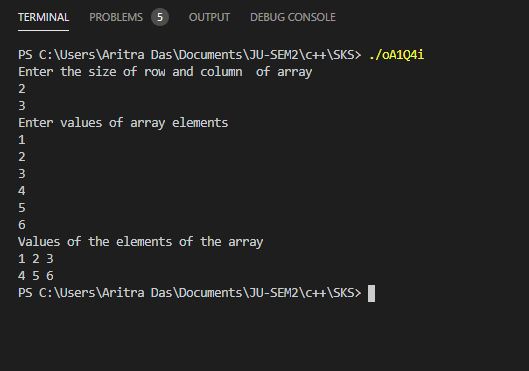
        for (int j = 0; j < n; ++j)

            printf("%d ", \*(\*(arr + i) + j));

        printf("\n");

    }

}



5.

#include <stdio.h>

#include <conio.h>

#include <string.h>

void arrange(char name[][51],int age[],int n)

{

   int i,j,temp=0;

   char t[51];

     for(i=0;i<n-1;i++)

     {

       for(j=i+1;j<n;j++)

       {

          if(age[i]>age[j])

          {

             temp=age[i];

             age[i]=age[j];

             age[j]=temp;

             strcpy(t,name[i]);

             strcpy(name[i],name[j]);

             strcpy(name[j],t);

          }

       }

     }

}

void display(char name[][51],int age[],int n)

{

  int i,j;

  printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

  printf("Age             Name\n");

  printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

    for(i=0;i<n;i++)

     printf("%d             %s",age[i],name[i]);

}

int main()

{

   int n,i;

   printf("Enter the number of names to be listed:  ");

   scanf("%d",&n);

   int age[n];

   char name[n][51];

     for(i=0;i<n;i++)

     {

        printf("Enter name: ");

        scanf("%s",&name[i]);

        printf("Enter age: ");

        scanf("%d",age[i]);

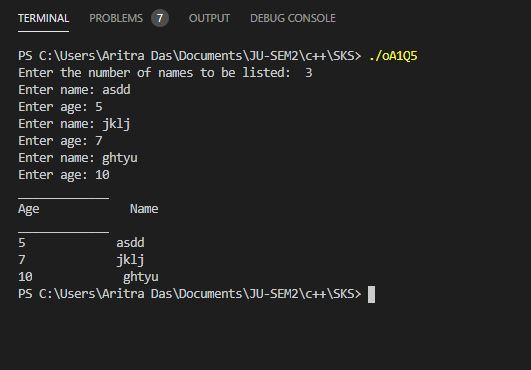
     }

     arrange(name,age,n);

     display(name,age,n);

    return 0;

}



6)

 #include<stdio.h>

struct student

{

    int roll\_no,marks;

    char name[25];

}stud[100],t;

int main()

{

    int i,j,n;

    printf("Enter the no of students\n");

    scanf("%d",&n);

    printf("enter student info as roll\_no , name , marks\n");

    for(i=0;i<n;i++)

    {

        scanf("%d %s %d",&stud[i].roll\_no,stud[i].name,&stud[i].marks);

    }

    for(i=0;i<n;i++)

    {

        for(j=0;j<n-1;j++)

        {

            if(stud[j].marks<stud[j+1].marks)

            {

                t=stud[j];

                stud[j]=stud[j+1];

                stud[j+1]=t;

            }

        }

    }

    printf("\nStudent info in terms of marks from highest to lowest\n");

    printf("\nROLL\_NO\t\tNAME\t\tMARKS\n");

    printf("\n");

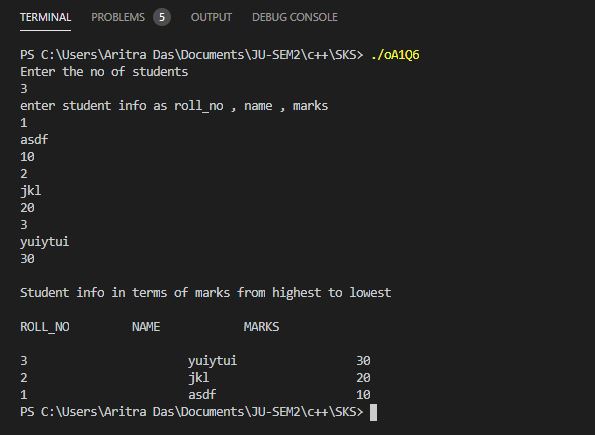
    for(i=0;i<n;i++)

    {

        printf("%d\t\t\t%s\t\t\t%d\n",stud[i].roll\_no,stud[i].name,stud[i].marks);

    }

}



7) #include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#include<string.h>

void Insert(char [], int , int );

void remove1(int roll);

void display();

int CheckEmpty();

struct Node{

 char name [20];

int roll;

int score;

struct Node \*next;//\*head=NULL;

};struct Node\* head=NULL;

int main()

{

 int rl,choice,scr;

 char nm[20];

 char c;

 do{

         printf("Enter 1 for insert 2 for remove 3 for display\n");

         scanf("%d",&choice);

 switch(choice)

 {

  case 1:

  {

   int x,roll;

   fflush(stdin);

   printf("Enter Value to be Insertedthat is Name, Roll  No, Score of the student\n");

   gets(nm);

   scanf("%d",&roll);

   scanf("%d",&scr);

   Insert(nm,roll,scr);

     break;

    case 2:

     printf("Enter roll to be removed starting from 1\n");

     scanf("%d",&rl);

     remove1(rl);

     break;

    case 3:

    display();

   break;

  }

  default:

  {

   printf("Not A Valid Choice\n");

   break;

  }

 }

 printf("Press y to continue else any letter\n");

 fflush(stdin);

 c=getche();

 printf("\n");

 }while(c=='Y' || c=='y');

 return(0);

}

void Insert(char nm[], int roll, int scr){

   struct Node \*newNumber;

   struct Node \*temp,\*temp1;

   int f=0;

   newNumber = (struct Node\*)malloc(sizeof(struct Node));

 if(head==NULL)

 {

 newNumber->roll = roll;

   newNumber->score=scr;

   strcpy(newNumber->name,nm);

   newNumber->next=NULL;

  head=newNumber;

 }

  else

  {

  temp=head;

  while(temp->next!=NULL)

  {

   temp=temp->next;

  }

   newNumber->roll = roll;

   newNumber->score=scr;

   strcpy(newNumber->name,nm);

   newNumber->next=NULL;

  temp->next=newNumber;

  newNumber=temp;

 }

}

void remove1(int roll){

 int flag=CheckEmpty();

 struct node \*temp1;

 if(flag==1)

 {

      struct Node \*temp,\*prev;

     temp=head;

      while(temp->next!=NULL)

     {

      if(temp->roll==roll)

      {

        prev->next=temp->next;

        temp=temp->next;

        break;

    }

      temp=temp->next;

      }

     prev=temp;

 }

else

{

  printf("List is empty\n");

 }

}

void display(){

 int flag=CheckEmpty();

 if(flag==1)

 {

      struct Node \*temp;

     temp=head;

      while(temp->next!=NULL)

     {

      printf("%d %s %d\n",temp->roll,temp->name,temp->score);

      temp=temp->next;

      }

      printf("%d %s %d",temp->roll,temp->name,temp->score);

        printf("\n");

 }

 else

 {

  printf("Not Available\n");

 }

}

int CheckEmpty()

{

 if(head==NULL)

 return 0;

 else

 return 1;

}

