

```
/*
Title- 1. Write a C program to to represent 1-D array using Dynamic Memory Allocation.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/
```

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    int n,i;
    int* arr=NULL;
    printf("Enter how many elements do you want to enter into the array: ");
    scanf("%d",&n);
    arr=(int*)malloc(n*sizeof(int));
    printf("enter elements into the array: \n");
    for(i=0;i<n;i++)
    {
        scanf("%d",arr+i);
    }
    printf("Entered array elements are: [");
    for(i=0;i<n;i++)
    {
        printf("%d, ",*(arr+i));
    }
    printf("]\n");
    free(arr);
    arr=NULL;
}
```

```
/*
Title- 2. Write a C program to sort 1-D array in ascending order using Dynamic Memory
Allocation.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/
```

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    int n,i,j,temp;
    int* arr=NULL;
    printf("Enter how many elements do you want to enter into the array: ");
    scanf("%d",&n);
    arr=(int*)malloc(n*sizeof(int));
    printf("enter array elements to sort: \n");
    for(i=0;i<n;i++)
    {
        scanf("%d",arr+i);
    }
    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(*(arr+j)<=*(arr+i))

```

```

        {
            temp=*(arr+j);
            *(arr+j)=*(arr+i);
            *(arr+i)=temp;
        }
    }
}
printf("Sorted array elements are: [");
for(i=0;i<n;i++)
{
    printf("%d, ",*(arr+i));
}
printf("]\n");
free(arr);
arr=NULL;
}

```

```

/*
Title- 3. Write a C program to search given element in 1-D array using binary search
method (Use Dynamic Memory Allocation to represent an array).
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

```

```

#include<stdio.h>
#include<stdlib.h>
void main()
{
    int n,i,j,temp,element,l,r,m,flag=0;
    int* arr=NULL;
    printf("Enter how many elements do you want to enter into the array: ");
    scanf("%d",&n);
    arr=(int*)malloc(n*sizeof(int));
    printf("enter array elements : \n");
    for(i=0;i<n;i++)
    {
        scanf("%d",arr+i);
    }
    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(*(arr+j)<=*(arr+i))
            {
                temp=*(arr+j);
                *(arr+j)=*(arr+i);
                *(arr+i)=temp;
            }
        }
    }
    printf("enter the element to search in array: ");
    scanf("%d",&element);
    l=0;
    r=n-1;

    do
    {
        m=(r+l)/2;
        if(*(arr+m)==element)

```

```

        {
            flag=1;
            break;
        }
        else
        {
            if(*(arr+m)>element)
            {
                r=m;
            }
            else
            {
                if(*(arr+m)<element)
                {
                    l=m;
                }
            }
        }
    }while(m!=0);
    if(flag==1)
    {
        printf("element found...\n");
    }
    else
    {
        printf("element not found...\n");
    }
    free(arr);
    arr=NULL;
}

```

```

/*
Title- 4. Write a C program to find second highest element in given 1-D array using
Dynamic Memory Allocation.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

```

```

#include<stdio.h>
#include<stdlib.h>
void main()
{
    int n,i,j,temp;
    int* arr=NULL;
    printf("Enter how many elements do you want to enter into the array: ");
    scanf("%d",&n);
    arr=(int*)malloc(n*sizeof(int));
    printf("enter array elements to sort: \n");
    for(i=0;i<n;i++)
    {
        scanf("%d",arr+i);
    }
    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(*(arr+j)>=*(arr+i))
            {

```

```

        temp=*(arr+j);
        *(arr+j)=*(arr+i);
        *(arr+i)=temp;
    }
}
printf("Second heighest element in array is: %d\n",*(arr+1));
free(arr);
arr=NULL;
}

```

```

/*
Title- 5. Write a C program to reverse an given 1-D without using sorting algorithms.
(Use Dynamic Memory Allocation to represent an array).
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

```

```

#include<stdio.h>
#include<stdlib.h>
void main()
{
    int n,i,j,temp;
    int* arr=NULL;
    printf("Enter how many elements do you want to enter into the array: ");
    scanf("%d",&n);
    arr=(int*)malloc(n*sizeof(int));
    printf("enter elements into the array: \n");
    for(i=0;i<n;i++)
    {
        scanf("%d",arr+i);
    }
    j=n-1;
    for(i=0;i<=j;i++,j--)
    {
        temp=*(arr+i);
        *(arr+i)=*(arr+j);
        *(arr+j)=temp;
    }

    printf("reversed array elements are: [");
    for(i=0;i<n;i++)
    {
        printf("%d, ",*(arr+i));
    }
    printf("]\n");
    free(arr);
    arr=NULL;
}

```

```

/*
Title- 6. Write a C program to represent 2-D array(Matrix) using Dynamic Memory
Allocation.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)

```

```

*/

#include<stdio.h>
#include<stdlib.h>
void main()
{
    int** ptr=NULL;
    int c,r,i,j;
    printf("How many columns Do you want in matrix: ");
    scanf("%d",&c);
    printf("How many Rows Do you want in matrix: ");
    scanf("%d",&r);
    ptr=(int**)malloc(r*sizeof(int*));
    for(i=0;i<c;i++)
    {
        *(ptr+i)=(int*)malloc(c*sizeof(int));
    }
    printf("enter the elements in the matrix: \n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            scanf("%d",(*(ptr+i)+j));
        }
    }
    printf("entered elements are: \n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            printf("%d\t",(*(ptr+i)+j));
        }
        printf("\n");
    }
    free(ptr);
    ptr=NULL;
}

```

```

/*
Title- 7. Write a C program to represent a matrix of Strings using Dynamic
Memory Allocation.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

```

```

#include<stdio.h>
#include<stdlib.h>
void main()
{
    char*** p=NULL;
    int i,j,len,r,c;
    char ch;
    printf("Enter th no.of rows you want: ");
    scanf("%d",&r);
    printf("Enter the no. of columns you want: ");
    scanf("%d",&c);
    p=(char***)malloc(r*sizeof(char*));
    for(i=0;i<r;i++)
    {
        *(p+i)=(char**)malloc(c*sizeof(char));
    }
}

```

```
}
printf("Enter the strings: \n");
for(i=0;i<r;i++)
{
    for(j=0;j<c;j++)
    {
        (*(p+i)+j)=(char*)malloc(sizeof(char));
        (*(p+i)+j+0)='\0';
        len=1;
        scanf(" ");
        do
        {
            scanf("%c",&ch);
            if(ch!='\n')
            {
                len++;
                (*(p+i)+j)=(char*)realloc(*(p+i)+j,len*sizeof(char));
                (*(p+i)+j+(len-2))=ch;
                (*(p+i)+j+(len-1))='\0';
            }
        }while(ch!='\n');
    }
}
printf("Entered matrix of strings is: \n");
for(i=0;i<r;i++)
{
    for(j=0;j<c;j++)
    {
        printf("%s\t",(*(p+i)+j));
    }
    printf("\n");
}
free(p);
p=NULL;
}
```