```c
/*
Title- 1. Write a C program to check all pointers size on different environments.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

#include<stdio.h>
void main()
{
    int* ptr=NULL;
    printf("Size of Pointer is: %ld bytes\n",sizeof(ptr));
}




/*
Title- 2. Write a C program to find out a multiplication matrix using Dynamic memory
allocation.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/


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

void main()
{
    int **p=NULL, **q=NULL, **r=NULL;
    int i,j,r1,c1,r2,c2,k;
    printf("enter no. of rows you want to enter into the array1: ");
    scanf("%d",&r1);
    printf("enter no. of columns you want to enter into the array1: ");
    scanf("%d",&c1);
    printf("enter no. of rows you want to enter into the array2: ");
    scanf("%d",&r2);
    printf("enter no. of columns you want to enter into the array2: ");
    scanf("%d",&c2);
    if(c1==r2)
    {

//         ----creating matrix 1 and adding elements in it ----
        p=(int**)malloc(r1*sizeof(int*));
        for(i=0;i<r1;i++)
        {
            *(p+i)=(int*)malloc(c1*sizeof(int));
        }
        printf("Enter the elements in the array1: \n");
        for(i=0;i<r1;i++)
        {
            for(j=0;j<c1;j++)
            {
                scanf("%d",(*(p+i)+j));
            }
        }
//     --------- creating matrix 2 and adding elements in it ----
         q=(int**)malloc(r2*sizeof(int*));
        for(i=0;i<r2;i++)
        {
            *(q+i)=(int*)malloc(c2*sizeof(int));
        }
        printf("Enter the elements in the array2: \n");
```

```
        for(i=0;i<r2;i++)
        {
            for(j=0;j<c2;j++)
            {
                scanf("%d",(*(q+i)+j));
            }
        }
//       -----  creating matrix 3 to store multiplication of 2 matrices -----
        r=(int**)malloc(r1*sizeof(int*));
        for(i=0;i<r1;i++)
        {
            *(r+i)=(int*)calloc(sizeof(int),c2);
        }
        for(i=0;i<r1;i++)
        {
            for(j=0;j<c2;j++)
            {
                for(k=0;k<c1;k++)
                {
                    (*(*(r+i)+j))+=(*(*(p+i)+k)) * (*(*(q+k)+j));
                }
            }
        }
        printf("Multiplication matrix is: \n");
        for(i=0;i<r1;i++)
        {
            for(j=0;j<c2;j++)
            {
                printf("%d\t",*(*(r+i)+j));
            }
            printf("\n");
        }
        free(p);
        free(q);
        free(r);
        p=q=r=NULL;

    }
    else
    {
        printf("Please enter proper matrix sizes....\n");
    }


}
```

```
/*
Title- 3. Write a c program to accept string from user and print String using character
pointer only (static memory allocation).
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/
#include<stdio.h>
void main()
{
    char str[100];
    char* p = str;
    int i=0;
    printf("Enter the string: ");
    fgets(str,sizeof(str),stdin);
```

```
    printf("Entered string is: ");
    for(i=0;*(p+i)!='\0' && *(p+i)!='\n';i++)
    {
        printf("%c",*(p+i));
    }
    printf("\n");
}
```

```
/*
Title- 4. Write a C program to print diagonal elements in matrix using Dynamic memory
allocation.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

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

void main()
{
    int** p=NULL;
    int r,c,i,j;
    printf("enter no. of rows you want in matrix: ");
    scanf("%d",&r);
    printf("Enter how many columns you want in matrix: ");
    scanf("%d",&c);
    p=(int**)malloc(r*sizeof(int*));
    for(i=0;i<r;i++)
    {
        *(p+i)=(int*)malloc(c*sizeof(int));
    }
    printf("enter the elements in matrix: \n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            scanf("%d",(*(p+i)+j));
        }
    }
    printf("Diagobal elements are: [");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            if(j==i)
            {
                printf("%d, ",*(*(p+i)+j));
            }
        }
    }
    printf("]\n");
    free(p);
    p=NULL;
}
```

```c
/*
Title- 5. Write a C program to demonstrate malloc , calloc , realloc and free function.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

#include<stdio.h>
#include<stdlib.h>
void main()
{
    int n,i,m;
    int *arr1=NULL,*arr2=NULL,*arr3=NULL;

    printf("Enter how many elements do you want to enter into the array: ");
    scanf("%d",&n);

//     ---------- malloc -----------------
    arr1=(int*)malloc(n*sizeof(int));
    printf("enter elements into the array1: \n");
    for(i=0;i<n;i++)
    {
        scanf("%d",arr1+i);
    }
    printf("Entered array1 elements are: [");
    for(i=0;i<n;i++)
    {
        printf("%d, ",*(arr1+i));
    }
    printf("]\n");

//     ------ calloc ---------
    arr2=(int*)calloc(sizeof(int), n);
    printf("enter elements into the array2: \n");
    for(i=0;i<n;i++)
    {
        scanf("%d",arr2+i);
    }
    printf("Entered array2 elements are: [");
    for(i=0;i<n;i++)
    {
        printf("%d, ",*(arr2+i));
    }
    printf("]\n");

//   -----  realloc -----------
    printf("Enter new size for array2: ");
    scanf("%d",&m);
    arr3=(int*)realloc(arr2, m*sizeof(int));
    arr2=NULL;
    printf("enter elements into the modified array2 again: \n");
    for(i=0;i<m;i++)
    {
        scanf("%d",arr3+i);
    }
    printf("Entered  new array2 elements are: [");
    for(i=0;i<m;i++)
    {
        printf("%d, ",*(arr3+i));
    }
    printf("]\n");
    free(arr1);
    free(arr2);
    free(arr3);
    arr1=arr2=arr3=NULL;
```

```
}
```

```
/*
Title- 6. Write a C program to print all armstrong numbers from given integer 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,num,j,count,mult,sum,value;
    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("Armstring elements are: [");
    for(i=0;i<n;i++)
    {
        sum=0;
        count=0;
        num=*(arr+i);
        // to find the no. of digits in a number
        while(num!=0)
        {
            num/=10;
            count++;
        }
        //  to find the armstrong value
        num=*(arr+i);
        while(num!=0)
        {
            value= num%10;
            mult=1;
            for(j=1;j<=count;j++)
            {
                mult*=value;
            }
            sum+=mult;
            num/=10;
        }
        //  to check the armstrong or not
        if(sum==*(arr+i))
        {
            printf("%d, ",*(arr+i));
        }
    }
    printf("]\n");
    free(arr);
    arr=NULL;

}
```

```c
/*
Title- 7. Write a C Program to print Transpose matrixof given matrix using Dynamic memory
allocation.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/
#include<stdio.h>
#include<stdlib.h>
void main()
{
    int** p=NULL;
    int r,c,i,j;
    printf("Enter no. of rows you want in matrix: ");
    scanf("%d",&r);
    printf("Enter no. of columns you want in matrix: ");
    scanf("%d",&c);
    p=(int**)malloc(r*sizeof(int*));
    for(i=0;i<r;i++)
    {
        *(p+i)=(int*)malloc(c*sizeof(int));
    }
    printf("Enter elements in the matrix: \n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            scanf("%d",(*(p+i)+j));
        }
    }
    printf("Original matrix is: \n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            printf("%d\t",(*(*(p+i)+j)));
        }
        printf("\n");
    }
    printf("Transformed matrix is: \n");
    for(i=0;i<c;i++)
    {
        for(j=0;j<r;j++)
        {
            printf("%d\t",(*(*(p+j)+i)));
        }
        printf("\n");
    }
    free(p);
    p=NULL;
}
```

```c
/*
Title- 8. Write a C program to check whether given string is pallindrome or not using
```

```c
Dynamic memory
allocation.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/
#include<stdio.h>
#include<stdlib.h>
void main()
{
    char* str;
    char c;
    int len=1,flag=0,i,j;
    str=(char*)malloc(sizeof(char));
    printf("Enter the string: ");
    do
    {
        scanf("%c",&c);
        if(c!='\n')
        {
            len++;
            str=(char*)realloc(str,len*sizeof(char));
            *(str+(len-2))=c;
            *(str+(len-1))='\0';
        }
    }while(c!='\n');
    for(i=0;*(str+i)!='\0' && *(str+i)!='\n';i++);
    i--;
    for(j=0;j<=i;j++,i--)
    {
        if(*(str+i)!=*(str+j))
        {
            flag=1;
            break;
        }
    }
    if(flag==0)
    {
        printf("string is palindrome..\n");
    }
    else
    {
        printf("string is not palindrome..\n");
    }
    free(str);
    str=NULL;
}
```

```c
/*
Title- 9. Write a C program to sort only those array elements which are less than 100
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))
            {
                if(*(arr+j)<100)
                {
                    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- 10. Write a C program to check whether matrix is upper triangular matrix or not
using Dynamic
memory allocation.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/


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

void main()
{
    int** p=NULL;
    int r,c,i,j,flag=0;
    printf("Enter the no. of rows you want: ");
    scanf("%d",&r);
    printf("Enter the no. of columns you want: ");
    scanf("%d",&c);
    p=(int**)malloc(r*sizeof(int*));
    for(i=0;i<r;i++)
    {
        *(p+i)=(int*)malloc(c*sizeof(int));
    }
    printf("Enter matrix elements: \n");
```

```
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            scanf("%d",(*(p+i)+j));
        }
    }
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            if(j<i)
            {
                if(*(*(p+i)+j)!=0)
                {
                    flag=1;
                    break;
                }
            }
        }
        if(flag==1)
        {
            break;
        }
    }
    if(flag==1)
    {
        printf("matrix is not upper triangular..\n");
    }
    else
    {
        printf("Matrix is upper Triangular...\n");
    }
    free(p);
    p=NULL;
}
```

```
/*
Title- 11. Write a C program to print all prime numbers from given array among given
range 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,element,num;
    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("Enter the range upto which you want to find elements are prime or not: ");
```

```
    scanf("%d",&element);
    printf("prime numbers are: [");
    for(i=0;i<n;i++)
    {
        num=*(arr+i);
        j=2;
        if(num<element)
        {
            while(j<=num)
            {
                if(num%j==0)
                {
                    if(num==j)
                    {
                        printf("%d, ",num);
                    }
                    else
                    {
                        break;
                    }
                }
                j++;
            }
        }

    }
    printf("]\n");
    free(arr);
    arr=NULL;
}
```

```
/*
Title- 12. Write a C program to print like (using Dynamic memory allocation)
fun gun run
sun mun dun
bun hun kun
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

#include<stdio.h>
#include<stdlib.h>
void main()
{
    char*** p=NULL;
    int i,j,r,c,len;
    char ch;
    printf("Enter no. of rows you want: ");
    scanf("%d",&r);
    printf("Enter 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*));
    }
    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');
        }
    }
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            printf("%s\t",(*(*(p+i)+j)));
        }
        printf("\n");
    }
    free(p);
    p=NULL;
}
```

```
/*
Title-13. Write a C program to demonstrate dangling pointer 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 *array=NULL;

    array=(int*)malloc(n*sizeof(int));
    printf("Enter how many elements do you want to enter into the array: ");
    scanf("%d",&n);
    printf("enter elements into the array: \n");
    for(i=0;i<n;i++)
    {
        scanf("%d",array+i);
    }
    printf("Entered array elements are: [");
    for(i=0;i<n;i++)
    {
        printf("%d, ",*(array+i));
    }
```

```
    printf("]\n");
    free(array);
    array=NULL;
}
```

```
/*
Title- 14. Write a C program to store and display multiple(10) strings in one 1-
dimensional array 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,n,len;
    char ch;
    printf("How many strings you wants to enter: ");
    scanf("%d",&n);
    p=(char**)malloc(n*sizeof(char*));
    for(i=0;i<n;i++)
    {
        *(p+i)=(char*)malloc(sizeof(char));
    }
    printf("Enter the strings: \n");
    for(i=0;i<n;i++)
    {
        *(*(p+i)+0)='\0';
        printf("Enter string %d: ",i+1);
        len=1;
        scanf(" ");
        do
        {
            scanf("%c",&ch);
            if(ch!='\n')
            {
                len++;
                *(p+i)=(char*)realloc(*(p+i) , len*sizeof(char));
                *(*(p+i)+(len-2))=ch;
                *(*(p+i)+(len-1))='\0';
            }

        }while(ch!='\n');
    }
    for(i=0;i<n;i++)
    {
        printf("%s\n",(*(p+i)));
    }
    free(p);
    p=NULL;
}
```

```c
/*
Title- 15. Write a C program to reverse a string using Dynamic memory allocation.
Eg:
Input string:
______India____is_____________my_______country_____________
Output string: aidnI_si_ym_yrtnuoc
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

#include<stdio.h>
#include<stdlib.h>
void main()
{
    char* str=NULL;
    char c,temp;
    int len=1,i,j,k;
    str=(char*)malloc(sizeof(char));
    printf("Enter the string: ");
    do
    {
        scanf("%c",&c);
        if(c!='\n')
        {
            len++;
            str=(char*)realloc(str,len*sizeof(char));
            *(str+(len-2))=c;
            *(str+(len-1))='\0';
        }
    }while(c!='\n');
    //     logic for swapping each word at its place & removing the extra spaces....
    for(j=0;*(str+j)!='\0';)
    {
        while(*(str+j)==32)
        {
            j++;
        }
        for(i=j;*(str+i)!=32 && *(str+i)!='\0';i++);
        i--;
        k=i;
        while(i>=j)
        {
            printf("%c",*(str+i));
            i--;
        }
        if(*str+(k+1)!='\0')
        {
            printf(" ");
        }
        j=k;
        j++;

    }
    printf("\n");
    free(str);
    str=NULL;


}
/*
Title- 16. Write a C program to print only those words from one string which are present
in given another
string using 1-D Dynamic memory allocation.
```

```c
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

#include<stdio.h>
#include<stdlib.h>
void main()
{
    char *str1=NULL, *str2=NULL, *str3=NULL, *str4=NULL;
    char c;
    int len,length,i,j,k,l,m,n,z,flag;
    str1=(char*)malloc(sizeof(char));
    len=1;
    printf("Enter the string1: ");
    do
    {
        scanf("%c",&c);
        if(c!='\n')
        {
            len++;
            str1=(char*)realloc(str1,len*sizeof(char));
            *(str1+(len-2))=c;
            *(str1+(len-1))='\0';
        }
    }while(c!='\n');


    str2=(char*)malloc(sizeof(char));
    len=1;
    printf("Enter the string2: ");
    do
    {
        scanf("%c",&c);
        if(c!='\n')
        {
            len++;
            str2=(char*)realloc(str2,len*sizeof(char));
            *(str2+(len-2))=c;
            *(str2+(len-1))='\0';
        }
    }while(c!='\n');


    str4=(char*)malloc(sizeof(char));

    str3=(char*)malloc(sizeof(char));
    printf("Common Words in both strings are: [");

    for(l=0;*(str2+l)!='\0';)
    {
//      ------copying words from string 2-----
        while(*(str2+l)==32)
        {
            l++;
        }

        for(m=l;*(str2+m)!=32 && *(str2+m)!='\0';m++);
        m--;
        z=m;
        length=1;
        while(l<=m)
        {
            length++;
            str4=(char*)realloc(str4,length*sizeof(char));
            *(str4+(length-2))=*(str2+l);
```

```
            *(str4+(length-1))='\0';
            l++;
        }
        l=z;
        l++;
//  -----  copying words from string 1 -----
        for(j=0;*(str1+j)!='\0';)
        {
            while(*(str1+j)==32)
            {
                j++;
            }
            for(i=j;*(str1+i)!=32 && *(str1+i)!='\0';i++);
            i--;
            k=i;
            len=1;
            while(j<=i)
            {
                len++;
                str3=(char*)realloc(str3,len*sizeof(char));
                *(str3+(len-2))=*(str1+j);
                *(str3+(len-1))='\0';
                j++;
            }
            j=k;
            j++;

//   ----- checking words are equals or not -----
            for(n=0;*(str3+n)!='\0' && *(str4+n)!='\0';n++)
            {
                if(*(str3+n)==*(str4+n))
                {
                    flag=1;
                }
                else
                {
                    flag=0;
                    break;
                }
            }
            if(*(str3+n)==*(str4+n))
            {
                flag=1;
            }
            if(flag==1)
            {
                printf("%s, ",str3);
            }

        }
    }
    printf("]\n");
    free(str1);
    free(str2);
    free(str3);
    free(str4);
    str1=str2=str3=str4=NULL;

}




/*
Title- 17. Write a C program to count repeated words in given string using Dynamic memory
```

```
allocation.
string using 1-D Dynamic memory allocation.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

#include<stdio.h>
#include<stdlib.h>
void main()
{
    char *str1=NULL, *str3=NULL, *str4=NULL;
    char c;
    int len,length,i,j,k,l,m,n,z,flag,count,countw=0,posi,posj;
    str1=(char*)malloc(sizeof(char));
    len=1;
    printf("Enter the string1: ");
    do
    {
        scanf("%c",&c);
        if(c!='\n')
        {
            len++;
            str1=(char*)realloc(str1,len*sizeof(char));
            *(str1+(len-2))=c;
            *(str1+(len-1))='\0';
        }
    }while(c!='\n');

    str4=(char*)malloc(sizeof(char));

    str3=(char*)malloc(sizeof(char));
    printf("number of repeated words in the string is: ");

    for(l=0;*(str1+l)!='\0';)
    {
//      ------copying word from string 1-----
        while(*(str1+l)==32)
        {
            l++;
        }

        for(m=l;*(str1+m)!=32 && *(str1+m)!='\0';m++);
        m--;
        z=m;
        length=1;
        posi=l;
        while(l<=m)
        {
            length++;
            str4=(char*)realloc(str4,length*sizeof(char));
            *(str4+(length-2))=*(str1+l);
            *(str4+(length-1))='\0';
            l++;
        }
        l=z;
        l++;
//  -----  copying words from string 1 -----
        count=0;
        for(j=0;*(str1+j)!='\0';)
        {
            while(*(str1+j)==32)
            {
                j++;
            }
            for(i=j;*(str1+i)!=32 && *(str1+i)!='\0';i++);
```

```
            i--;
            k=i;
            len=1;
            posj=j;
            while(j<=i)
            {
                len++;
                str3=(char*)realloc(str3,len*sizeof(char));
                *(str3+(len-2))=*(str1+j);
                *(str3+(len-1))='\0';
                j++;
            }
            j=k;
            j++;

//   ----- checking words are equals or not -----
            for(n=0;*(str3+n)!='\0' && *(str4+n)!='\0';n++)
            {
                if(*(str3+n)==*(str4+n))
                {
                    flag=1;
                }
                else
                {
                    flag=0;
                    break;
                }
            }
            if(*(str3+n)==*(str4+n))
            {
                flag=1;
            }
            if(flag==1)
            {
                if(posi<=posj)
                {
                    count++;
                }
                else
                {
                    break;
                }
            }

        }
        if(count>1)
        {
            countw++;
        }

    }
    printf("%d\n",countw);
    free(str1);
    free(str3);
    free(str4);
    str1=str3=str4=NULL;

}
```

```c
/*
Title- 18. Write a C program to check whether matrix is upper triangular matrix or not
using Dynamic
memory allocation.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/


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

void main()
{
    int** p=NULL;
    int r,c,i,j,flag=0;
    printf("Enter the no. of rows you want: ");
    scanf("%d",&r);
    printf("Enter the no. of columns you want: ");
    scanf("%d",&c);
    p=(int**)malloc(r*sizeof(int*));
    for(i=0;i<r;i++)
    {
        *(p+i)=(int*)malloc(c*sizeof(int));
    }
    printf("Enter matrix elements: \n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            scanf("%d",(*(p+i)+j));
        }
    }
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            if(j<i)
            {
                if(*(*(p+i)+j)!=0)
                {
                    flag=1;
                    break;
                }
            }
        }
        if(flag==1)
        {
            break;
        }
    }
    if(flag==1)
    {
        printf("matrix is not upper triangular..\n");
    }
    else
    {
        printf("Matrix is upper Triangular...\n");
    }
    free(p);
    p=NULL;
}
```

```c
/*
Title- 19. Write a C program to find all negative numbers from given 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("Negative array elements are: [");
    for(i=0;i<n;i++)
    {
        if(*(arr+i)<0)
        {
            printf("%d, ",*(arr+i));
        }
    }
    printf("]\n");
    free(arr);
    arr=NULL;

}
```

```c
/*
Title- 20. Write a C program to sort array in descending 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);
    }
```

```c
    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;

}
```

```c
/*
Title- 21. Write a C program to print all numbers from array which are divisible by 5 and
7 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("array elements which are divisible by 5 & 7 are: [");
    for(i=0;i<n;i++)
    {
        if(*(arr+i)%5==0 || *(arr+i)%7==0)
        {
            printf("%d, ",*(arr+i));
        }
    }
    printf("]\n");
    free(arr);
    arr=NULL;

}
```