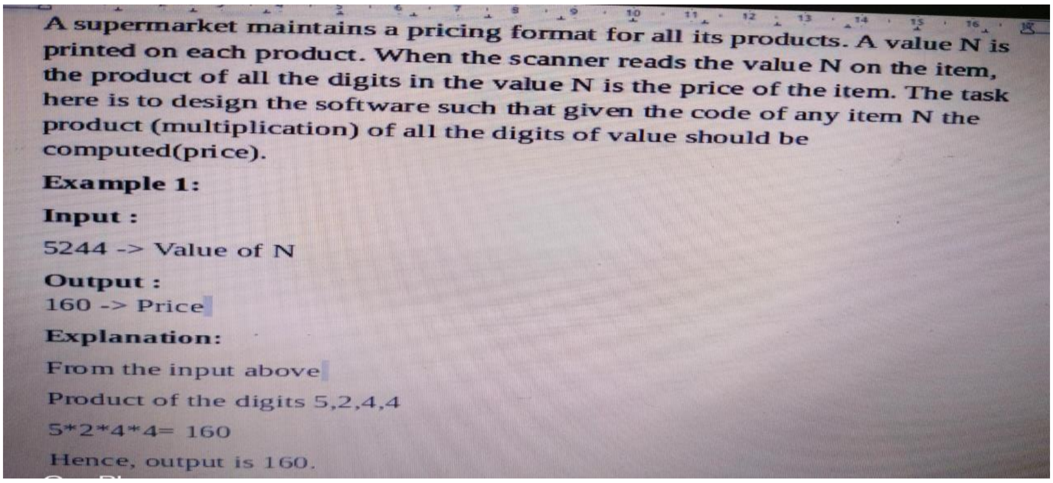
M1 Coding Practice

1. 

A supermarket maintains a pricing format for all its products. A value N is printed on each product. When the scanner reads the value N on the item, the product of all the digits in the value N is the price of the item. The task here is to design the software such that given the code of any item N the product (multiplication) of all the digits of value should be computed(price).

Example 1:

Input :

5244 -> Value of N

Output :

160 -> Price

Explanation:

From the input above

Product of the digits 5,2,4,4

5*2*4*4= 160

Hence, output is 160.

Vi product_digits.c

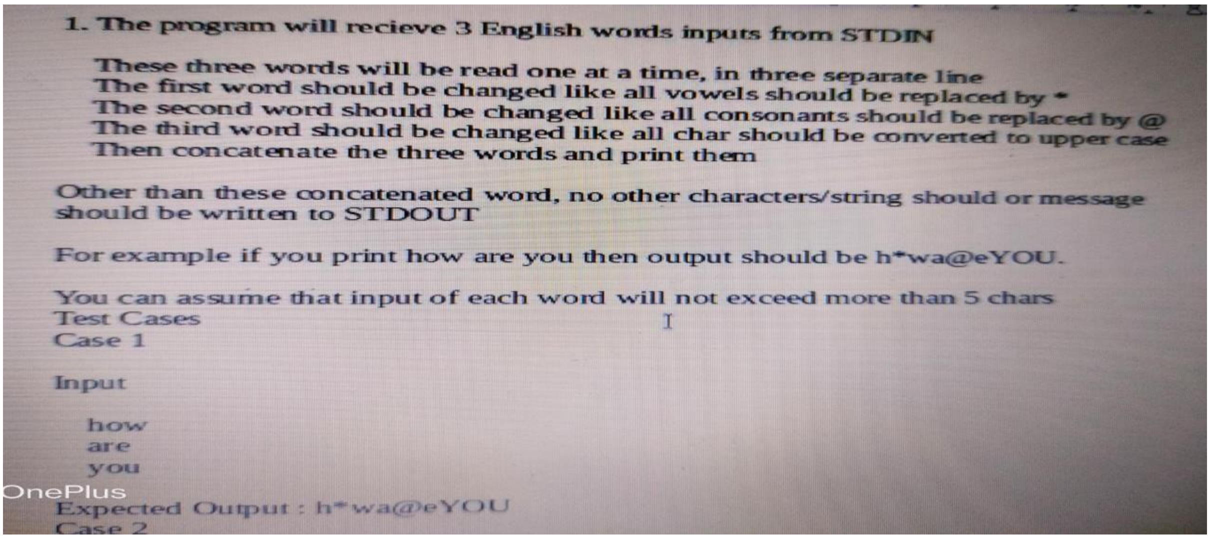
```
#include<stdio.h>
int main()
{
   int n,rem,prod=1;
   printf("Enter a number: ");
   scanf("%d",&n);
   while(n!=0)
   {
      rem=n%10;
       prod=prod*rem;
       n=n/10;
    }
    printf("%d\n",prod);
    return 0;
}
```

INPUT:5244

OUTPUT:160

Explanation:5*2*4*4=160

2.



1. The program will recieve 3 English words inputs from STDIN

These three words will be read one at a time, in three separate line
The first word should be changed like all vowels should be replaced by *
The second word should be changed like all consonants should be replaced by @
The third word should be changed like all char should be converted to upper case
Then concatenate the three words and print them

Other than these concatenated word, no other characters/string should or message should be written to STDOUT

For example if you print how are you then output should be h*wa@eYOU.

You can assume that input of each word will not exceed more than 5 chars
Test Cases
Case 1

Input

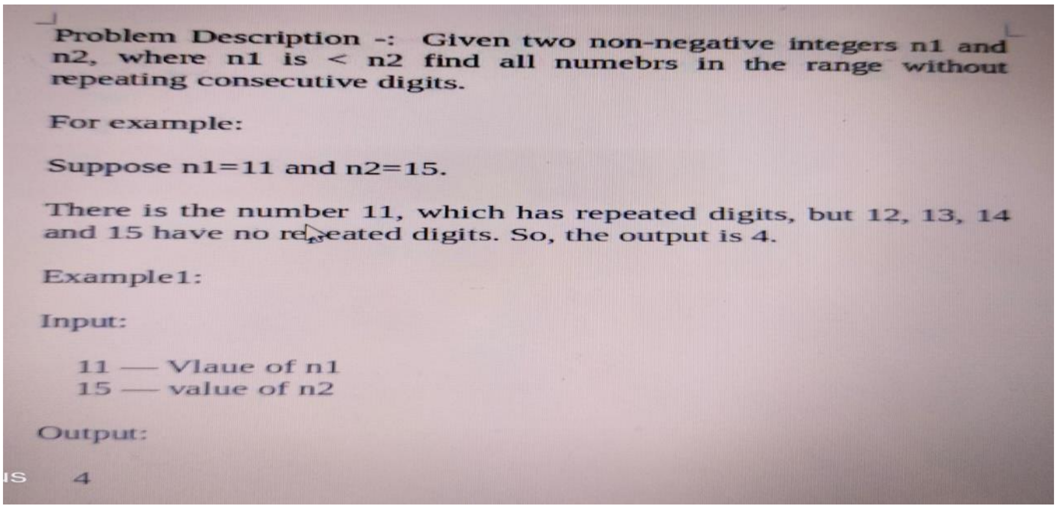
how
are
you

Expected Output : h*wa@eYOU
Case 2

Vi English_words.c

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

int main()
{
        int i;
        char a[100],b[100],c[100];

        scanf("%s",a);
        scanf("%s",b);
        scanf("%s",c);

        for(i=0;a[i]!='\0';i++)
        {
if(a[i]=='a'||a[i]=='e'||a[i]=='i'||a[i]=='o'||a[i]=='u'||a[i]=='A'||a[i]=='E'||a[i]=='I'||a[i]=='O'||a[i]=='U')
                        a[i]='*';
        }

        for(i=0;b[i]!='\0';i++)
        {
                if((b[i]>='a'&&b[i]<='z') || (b[i]>='A'&&b[i]<='Z'))
if(!(b[i]=='a'||b[i]=='e'||b[i]=='i'||b[i]=='o'||b[i]=='u'||b[i]=='A'||b[i]=='E'||b[i]=='I'||b[i]=='O'||b[i]=='U'))
                                b[i]='@';
        }
        for(i=0;c[i]!='\0';i++)
        {
              if(c[i]>='a'&&c[i]<='z')
                    c[i]=c[i]-32;
        }

        printf("%s%s%s\n",a,b,c);
        return 0;
}
```

```
INPUT:  how
         are
        you
OUTPUT: h*wa@eYOU
```



Problem Description -: Given two non-negative integers n1 and n2, where n1 is < n2 find all numebrs in the range without repeating consecutive digits.

For example:

Suppose n1=11 and n2=15.

There is the number 11, which has repeated digits, but 12, 13, 14 and 15 have no repeated digits. So, the output is 4.

Example1:

Input:

```
11 — Vlaue of n1
15 — value of n2
```

Output:

```
4
```

3.

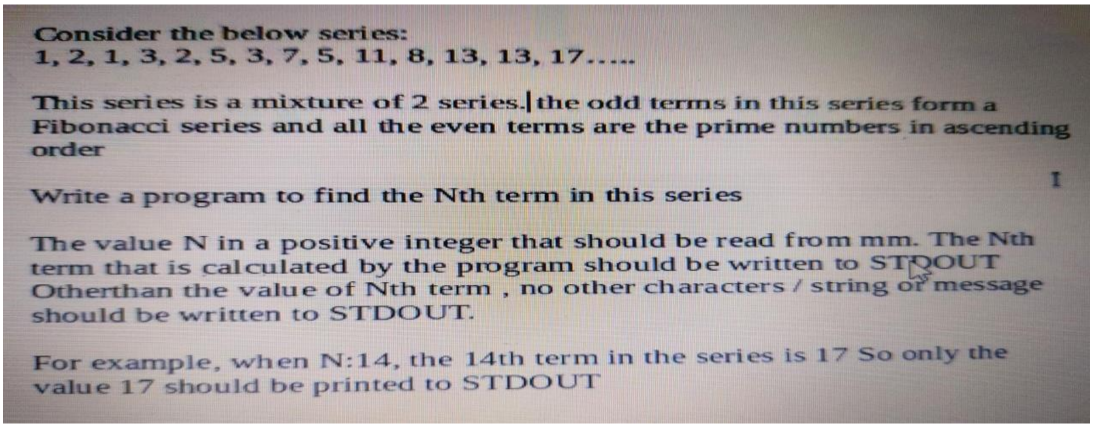
Vi non_neg_integers.c

```
#include<stdio.h>
int main()
{
       int n1,n2;
       int check1,check2,check3;
       printf("Enter two numbers n1 and n2\n");
       scanf("%d %d",&n1,&n2);
       for(int i=n1;i<=n2;i++)
       {
              check1=i;
              check2=check1%10;
              check1=check1/10;
              check3=check1%10;
              if(check2==check3)
              {
                    continue;
```

```
        }
        else if(check2!=check3)
        {
            printf(" %d ",i);
        }
        printf("\n");
    }
}
```

```
INPUT: Enter two numbers:
OUTPUT:- 11  15
            12 13 14 15
```

4.



Consider the below series:
1, 2, 1, 3, 2, 5, 3, 7, 5, 11, 8, 13, 13, 17.....

This series is a mixture of 2 series. the odd terms in this series form a Fibonacci series and all the even terms are the prime numbers in ascending order

Write a program to find the Nth term in this series

The value N in a positive integer that should be read from mm. The Nth term that is calculated by the program should be written to STDOUT Otherthan the value of Nth term , no other characters / string or message should be written to STDOUT.

For example, when N:14, the 14th term in the series is 17 So only the value 17 should be printed to STDOUT

Vi fib_prime.c

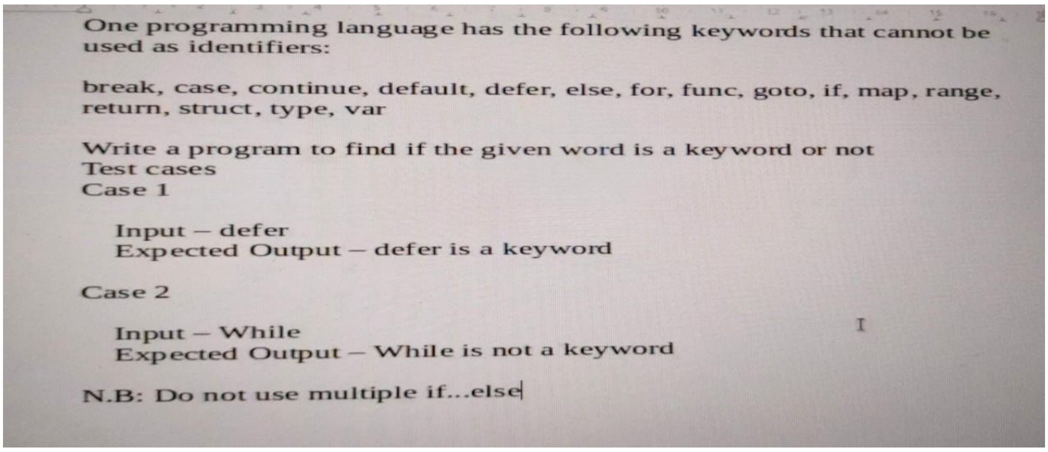
```
#include<stdio.h>
#define MAX 1000
void fibonacci(int n)
{
 int i, term1 = 0, term2 = 1, nextTerm;
 for (i = 1; i<=n; i++)
 {
  nextTerm = term1 + term2;
  term1 = term2;
  term2 = nextTerm;
 }
 printf("%d", term1);
}

void prime(int n)
{
 int i, j, flag, count =0;
 for (i=2; i<=MAX; i++)
 {
  flag = 0;
  for (j=2; j<i; j++)
  {
   if(i%j == 0)
   {
    flag = 1;
    break;
   }
  }
  if (flag == 0)
   count++;
  if(count == n)
  {
   printf("%d", i);
   break;
  }
 }
}
int main( )
{
 int n;
 scanf("%d", &n);
 if(n%2 == 1)
  fibonacci (n/2 + 1);
 else
  prime(n/2);
 return 0;
}
```

INPUT: 5

OUTPUT: 2

Explanation: 1   2   1   3   2   5   3   7   5 11   8   13   13   17……..

5.



One programming language has the following keywords that cannot be used as identifiers:

break, case, continue, default, defer, else, for, func, goto, if, map, range, return, struct, type, var

Write a program to find if the given word is a keyword or not

Test cases

Case 1

Input – defer

Expected Output – defer is a keyword

Case 2

Input – While

Expected Output – While is not a keyword

N.B: Do not use multiple if...else

Vi keywords.c

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

int main()
{
    char str[16][10]={"break","case","continue","default","defer","else","for","func","goto","if","map","range","return","struct","type","var"};
    char a[20];
    int f=0;
    scanf("%d",a);
    for(int i=0;i<16;i++)
    {
        if(strcmp(a,str[i])==0)
        {
            f=1;
            break;
        }
    }
    if(f==1)
    {
        printf("%s is a keyword",a);
    }
    else
    {
        printf("%s is not a keyword",a);
    }
}
```

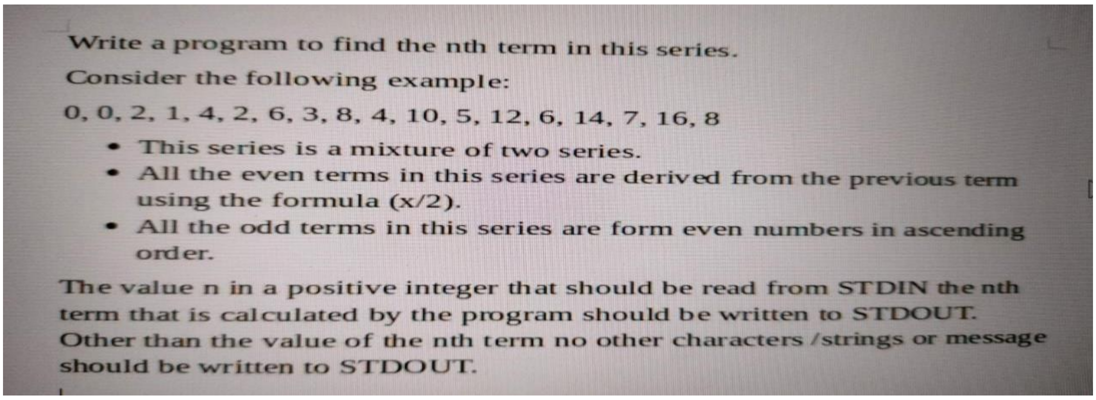
INPUT: while

OUTPUT: while is not a keyword

INPUT: defer

OUTPUT: defer is a keyword

6.



Write a program to find the nth term in this series.

Consider the following example:

0, 0, 2, 1, 4, 2, 6, 3, 8, 4, 10, 5, 12, 6, 14, 7, 16, 8

- This series is a mixture of two series.
- All the even terms in this series are derived from the previous term using the formula (x/2).
- All the odd terms in this series are form even numbers in ascending order.

The value n in a positive integer that should be read from STDIN the nth term that is calculated by the program should be written to STDOUT. Other than the value of the nth term no other characters /strings or message should be written to STDOUT.

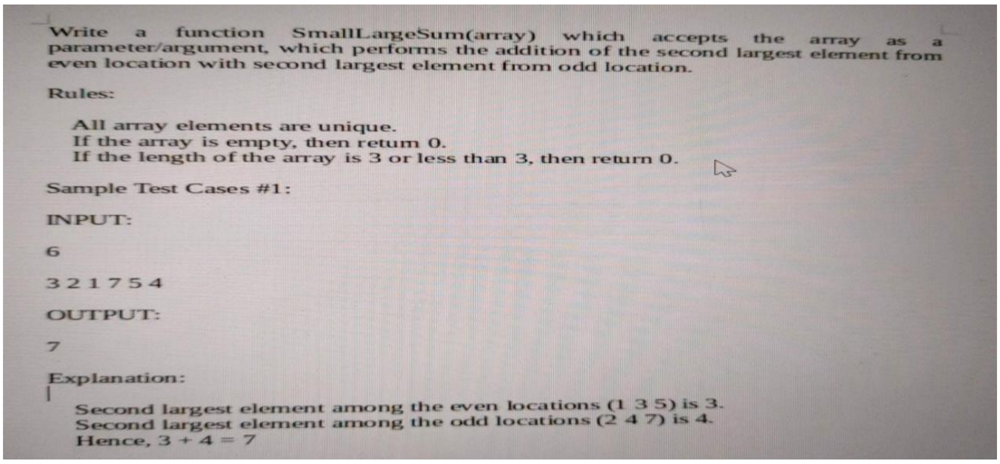
Vi even_odd.c

```
#include<stdio.h>
int main()
{
    int i,n,x=0,y=0;
    printf("Enter number: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
       if((i%2!=0)&&(i>1))
        {
              x=x+2;
        }
        else
        {
           y=x/2;
        }
     }

     if(n%2!=0)
     {
        printf("%d\n",x);
     }
     else
     {
        printf("%d\n",y);
     }
     printf("\n");
     return 0;
}
```

INPUT: 5

OUTPUT: 2

Explanation: 0   0   2 1   4 2   6   3   8   4   10   5   12   6 14   7   16   8……………………

7. 

Write a function SmallLargeSum(array) which accepts the array as a parameter/argument, which performs the addition of the second largest element from even location with second largest element from odd location.

Rules:

All array elements are unique.
If the array is empty, then return 0.
If the length of the array is 3 or less than 3, then return 0.

Sample Test Cases #1:

INPUT:

6

3 2 1 7 5 4

OUTPUT:

7

Explanation:

Second largest element among the even locations (1 3 5) is 3.
Second largest element among the odd locations (2 4 7) is 4.
Hence, 3 + 4 = 7

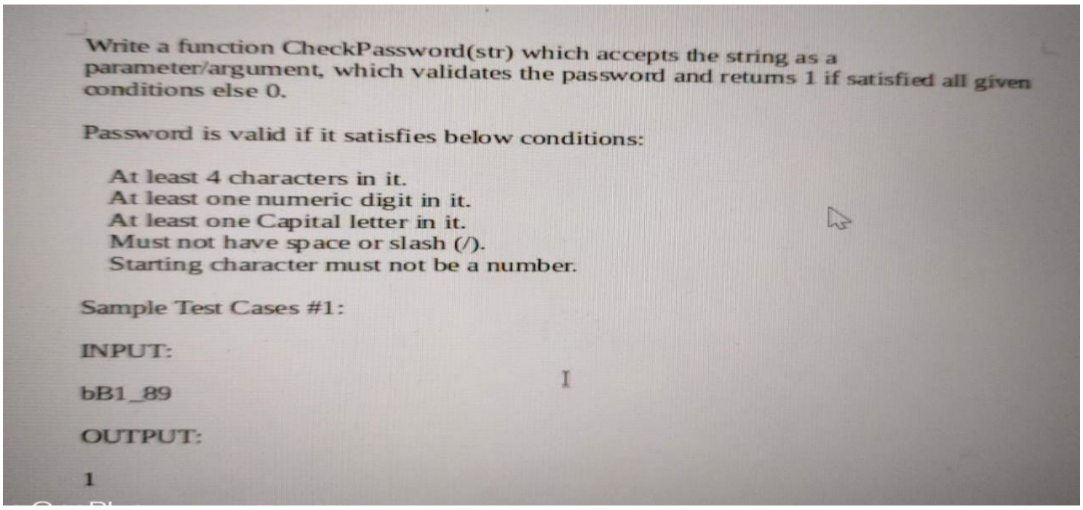
Vi second_large.c

```
#include <stdio.h>
int largeSmallSum(int *array, int n)
{
    int answer,i,j,temp;;
    int even[n],odd[n];
    int evencount=0,oddcount=0;
    if(n<=3)
    {
       answer=0;
    }
    else
    {
       even[0]=array[0];
       evencount=1;
       for(i=1;i<n;i++)
       {
          if(i%2==0)
          {
             even[evencount]=array[i];
             evencount++;
          }
          else
          {
             odd[oddcount]=array[i];
             oddcount++;
          }
       }
       for(i=0;i<evencount;i++)
       {
          for(j=i+1;j<evencount;j++)
          {
             if(even[i]>even[j])
```

```
                {
                    temp=even[i];
                    even[i]=even[j];
                    even[j]=temp;
                }
            }
        }
        for(i=0;i<oddcount;i++)
        {
            for(j=i+1;j<oddcount;j++)
            {
                if(odd[i]>odd[j])
                {
                    temp=odd[i];
                    odd[i]=odd[j];
                    odd[j]=temp;
                }
            }
        }
        answer=even[evencount-2]+odd[1];
    }
    return answer;
}

int main()
{
    int n,result,i;
    scanf("%d",&n);
    int array[n];
    for(i=0;i<n;i++)
    {
        scanf("%d",&array[i]);
    }
    result=largeSmallSum(array,n);
    printf("%d\n",result);
    return 0;
}
```

```
INPUT: 6
       3  2  1  7  5  4
OUTPUT: 7
```

8.



Write a function CheckPassword(str) which accepts the string as a parameter/argument, which validates the password and returns 1 if satisfied all given conditions else 0.

Password is valid if it satisfies below conditions:

At least 4 characters in it.
At least one numeric digit in it.
At least one Capital letter in it.
Must not have space or slash (/).
Starting character must not be a number.

Sample Test Cases #1:

INPUT:

bB1_89

OUTPUT:

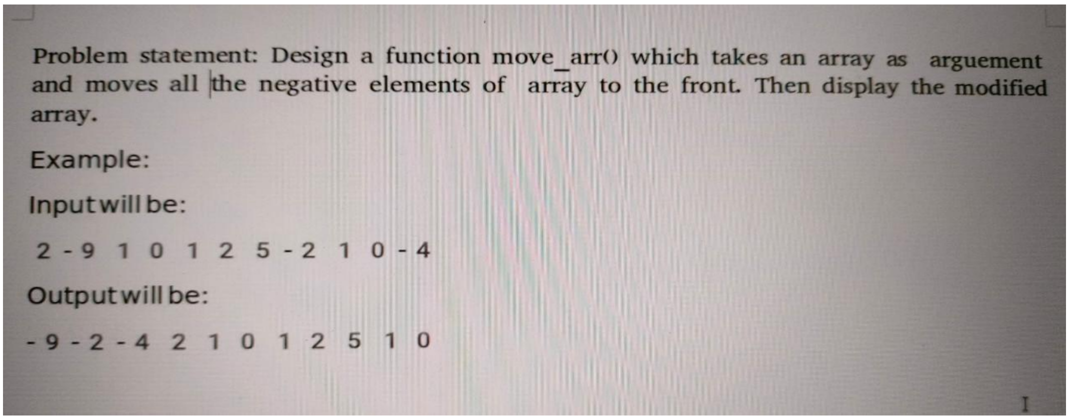
1

Vi password.c

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
int checkpassword(char *s)
{
       if(strlen(s)<5||isdigit(s[0]))
              return 0;
       int f1,f2;
       for(int i=0;i<strlen(s)-1;i++)
        {
              if(s[i]=='/'||s[i]=='0')
                     return 0;
              if(isupper(s[i]))
                     f1=1;
              if(isdigit(s[i]))
                     f2=1;
        }
        if(f1==1&&f2==1)
              return 1;
        else
              return 0;
}
int main()
{
        char str[20];
        printf("Enter the password:");
        fgets(str,19,stdin);
        if(checkpassword(str))
              printf("All condition satisfied\n");
        else
```

```
            printf("Incorrect format\n");
}
```

INPUT: Enter password: bB1_89
OUTPUT: all conditions satisfied

9. 

Problem statement: Design a function move_arr() which takes an array as arguement and moves all the negative elements of array to the front. Then display the modified array.

Example:

Input will be:

```
2 -9 1 0 1 2 5 -2 1 0 -4
```

Output will be:

```
-9 -2 -4 2 1 0 1 2 5 1 0
```

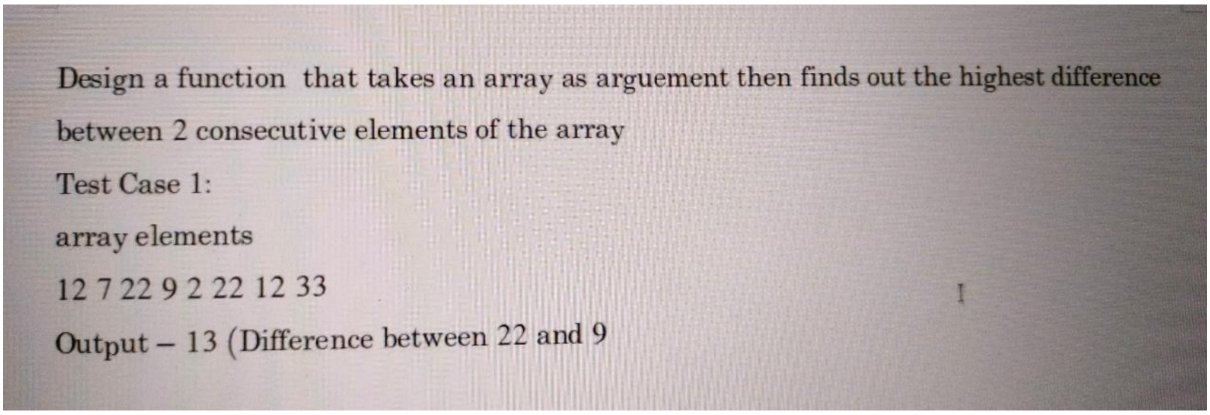
vi neg_numbers.c

```
#include<stdio.h>
int main()
{
      int arr[]={-9,2,1,0,12,5,-2,1,0,-4};
      int size=sizeof(arr)/sizeof(int);
      int j=0;
      for(int i=0;i<size;i++)
      {
            printf("%d ",arr[i]);
       }
       for(int i=0;i<size;i++)
       {
             if(arr[i]<0)
             {
                   int temp=arr[j];
                   arr[j]=arr[i];
                   arr[i]=temp;
                   j++;
             }
       }
       printf("\n");
       for(int i=0;i<size;i++)
       {
             printf("%d ",arr[i]);
       }
       printf("\n");
}
```

INPUT: 2 -9 1 0 1 2 5 -2 1 0 -4

OUTPUT: -9 -2 -4 2 1 0 1 2 5 1 0

10.



Design a function that takes an array as arguement then finds out the highest difference between 2 consecutive elements of the array

Test Case 1:

array elements

12 7 22 9 2 22 12 33

Output – 13 (Difference between 22 and 9

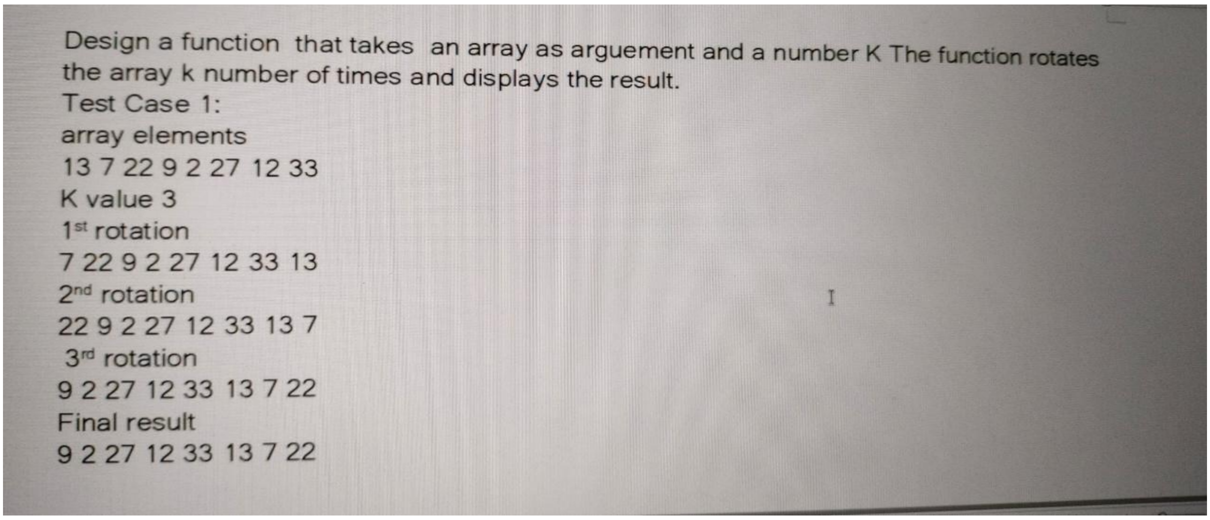
Vi high_diff.c

```
#include<stdio.h>

int diff_no(int arr[],int size);
int main()
{
        int arr[]={12,7,22,9,2,22,12,33};
        int size=sizeof(arr)/sizeof(int);
        diff_no(arr,size);
}

int diff_no(int arr[],int size)
{
        int diff=0;diff1=0;
        diff1=arr[0]-arr[1];
        for(int i=1;i<size;i++)
        {
                diff=arr[i]-arr[i+1];
                if(diff1<diff)
                {
                        diff1=diff;
                }
        }
        printf("\n %d \n",diff1);
}
```

INPUT: 12 7 22 9 2 22 12 33

OUTPUT: 13 (diff b/w 22 &9)

11.



Design a function that takes an array as arguement and a number K The function rotates the array k number of times and displays the result.

Test Case 1:

array elements

13 7 22 9 2 27 12 33

K value 3

1st rotation

7 22 9 2 27 12 33 13

2nd rotation

22 9 2 27 12 33 13 7

3rd rotation

9 2 27 12 33 13 7 22

Final result

9 2 27 12 33 13 7 22

Vi rotate_arr.c

```
#include <stdio.h>
#include <stdlib.h>
void rotate_array(int a[],int length,int k)
{
    for(int i = 0; i < k; i++)
    {
        int j,first;
        first = a[0];

         for(j = 0; j < length-1; j++)
         {
             a[j] = a[j+1];
         }
         a[j] = first;
    }

    printf("\n");

    printf("Array after left rotation: \n");
    for(int i = 0; i < length; i++){
        printf("%d ", a[i]);
    }
}

int main()

{
    int arr[] = {13,7,22,9,2,27,12,33};

    int size = sizeof(arr)/sizeof(arr[0]);
    int n;
    scanf("%d",&n);

    printf("Original array: \n");
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    rotate_array(arr,size,n);
    printf("\n");
    return 0;
```

INPUT: 13 7 22 9 2 27 12 33

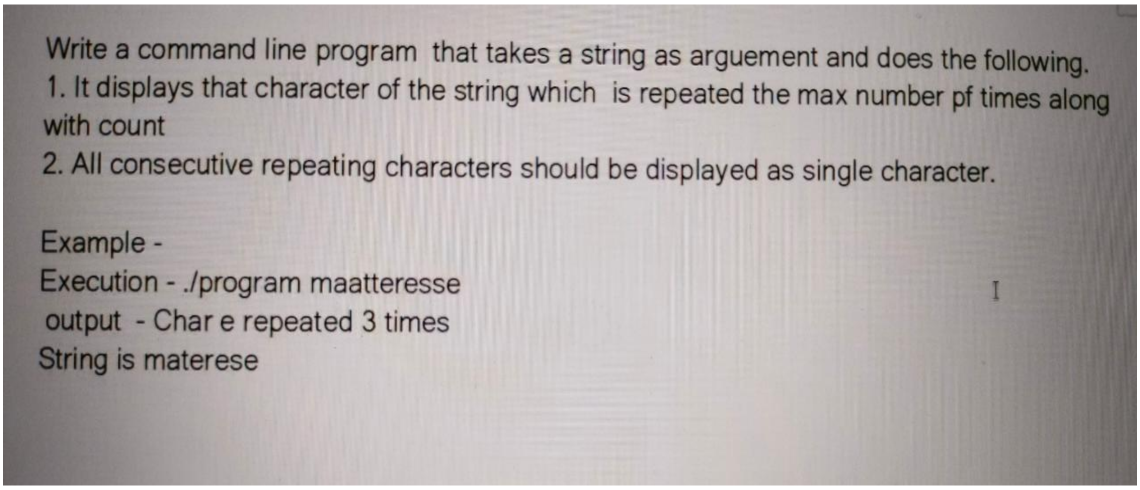
OUTPUT: Original array:

13 7 22 9 2 27 12 33

Array after left rotation:

27 12 33 13 7 22 9 2

12.



Write a command line program that takes a string as arguement and does the following.

1. It displays that character of the string which is repeated the max number pf times along with count
2. All consecutive repeating characters should be displayed as single character.

Example -

Execution - ./program maatteresse

output - Char e repeated 3 times

String is materese

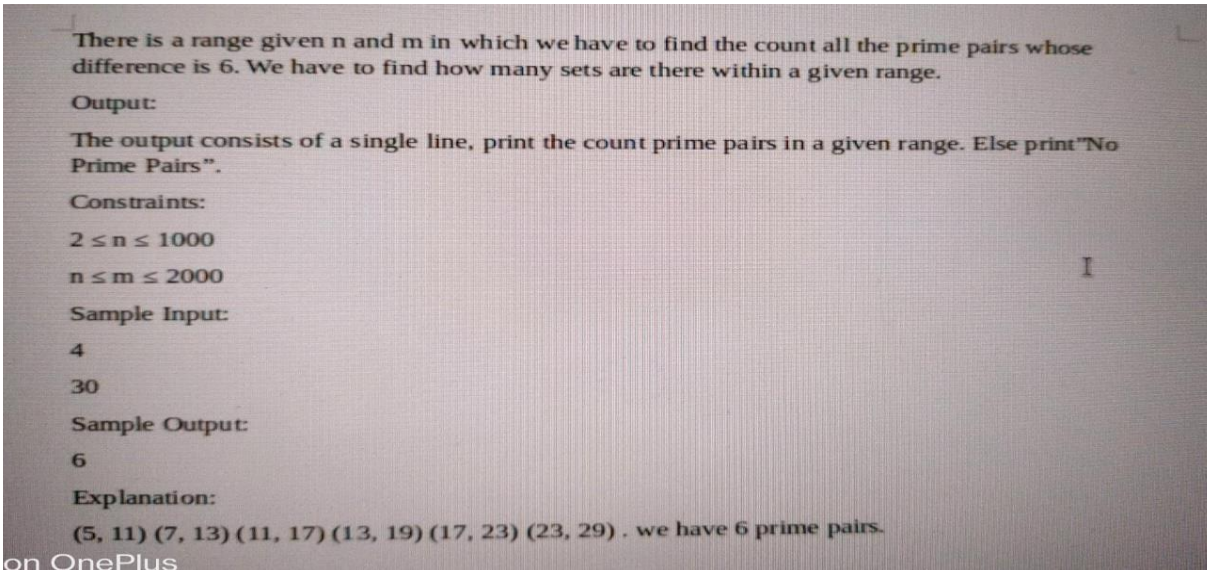
Vi cmd_line1.c

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main(int args_count,char *v[])
{
    if(args_count!=2)
    {
        printf("Invalid input\n");
        return EXIT_FAILURE;
    }
    char s[40],w;
    strcpy(s,v[1]);
    int len=strlen(s),f,c,max=1;
    for(int i=0;i<len;i++)
    {
        c=0;
        for(int j=i+1;j,len;j++)
        {
            if(s[i]==s[j])
            {
                for(int k=j;k<len;k++)
                {
                    s[k]=s[k+1];
                }
                s[len-1]='\0';
                len--;
                j--;
                c++;
            }
        }
        if(max<c)
        {
            max=c;
            w=s[i];
        }
    }
    printf("%s\n",s);
    printf("max:%d;char:%c\n",max,w);
}
```

INPUT: maatteresse

OUTPUT: max:12;char:3

13.



There is a range given n and m in which we have to find the count all the prime pairs whose difference is 6. We have to find how many sets are there within a given range.

Output:

The output consists of a single line, print the count prime pairs in a given range. Else print"No Prime Pairs".

Constraints:

2 ≤ n ≤ 1000

n ≤ m ≤ 2000

Sample Input:

4

30

Sample Output:

6

Explanation:

(5, 11) (7, 13) (11, 17) (13, 19) (17, 23) (23, 29) . we have 6 prime pairs.

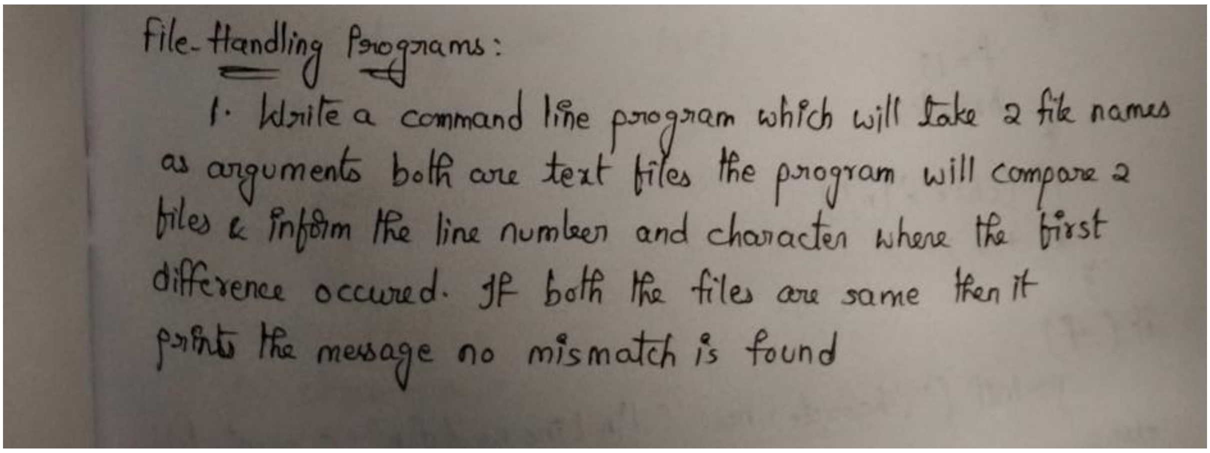
Vi prime_count.c

```
enter range:4
30
(5,11)(7,13)(11,17)(13,19)(17,23)(23,29)
we have 6 numbers with difference 6
```

```
#include<stdio.h>
int check_prime(int n)
{
        for(int i=2;i<n;i++)
        {
                if(n%i==0)
                {
                        return 0;
                }

        }
        return 1;
}

int main()
{
        int n,m,c=0;
        while(1)
        {
                printf("enter range:");
                scanf("%d %d",&n,&m);
                if(n<2 ||  n>1000)
                {
                        printf("enter n should be in the range\n");
                continue;
            }
            if(m<n || m>2000)
            {
                printf("enter m should be in the range\n");
                continue;

            }
            break;
        }
        for(int i=n;i<m;i++)
        {
            if(check_prime(i) && check_prime(i+6))
            {
                printf("(%d,%d)",i,i+6);
                c=c+1;
            }
        }
        printf("\nwe have %d numbers with difference 6\n",c);
}
```

```
INPUT: enter range:4
                   30
OUTPUT: (5,11)(7,13)(11,17)(13,19)(17,23)(23,29)
          we have 6 numbers with difference 6
```

14.



File-Handling Programs:

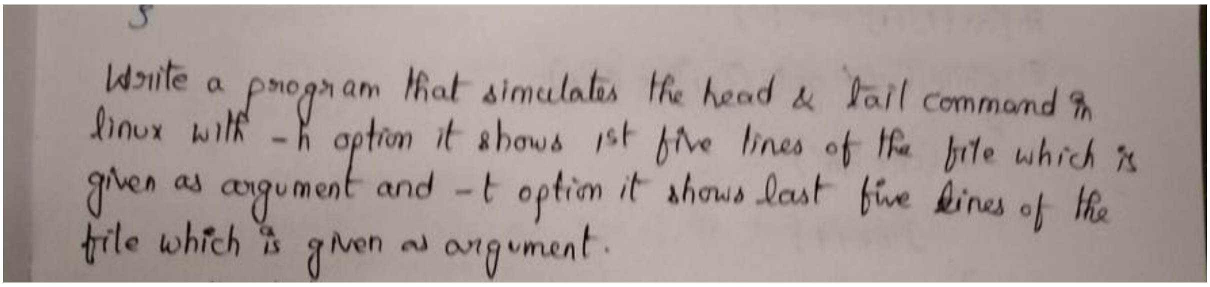
1. Write a command line program which will take 2 file names as arguments both are text files the program will compare 2 files & inform the line number and character where the first difference occured. If both the files are same then it prints the message no mismatch is found

vi comp.c

```
#include <stdio.h>
int main(int c, char **v)
{
      FILE *fp1,*fp2;
```

```
    fp1=fopen(v[1],"rb");
    fp2=fopen(v[2],"rb");

    char ch1,ch2;
    int lctr=1;
    int cctr=0;
    while(((ch1=getc(fp1))!=EOF) && ((ch2=getc(fp2))!=EOF))
    {
        cctr++;
        if(ch1=='\n' || ch2=='\n')
            lctr++;
        if(ch1!=ch2)
        {
            break;
        }

    }
    if((ch1!=EOF) && (ch2!=EOF))
    {
        printf("\nFirst differerence at line %d char %d\n",lctr,cctr);
    }
    fclose(fp1);
    fclose(fp2);
}
```

15.



5

Write a program that simulates the head & tail command in linux with -h option it shows 1st five lines of the file which is given as argument and -t option it shows last five lines of the file which is given as argument.

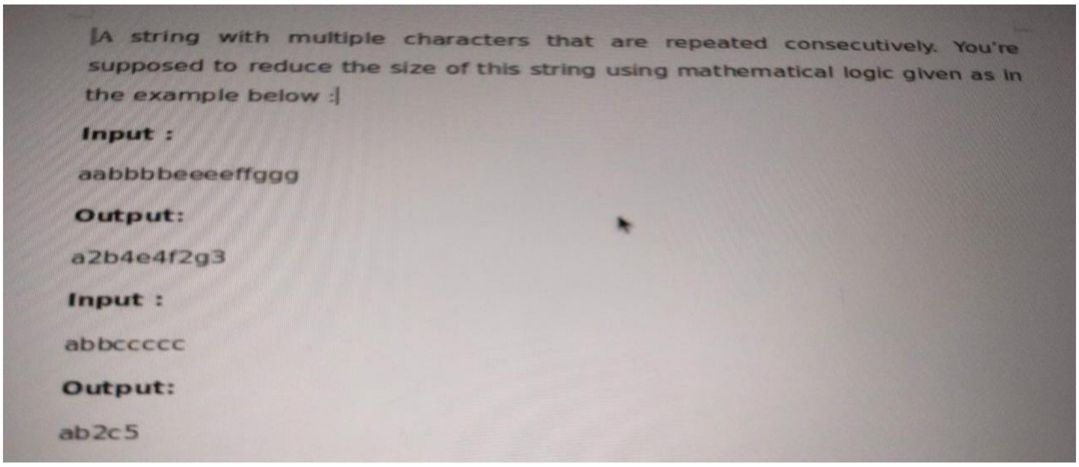
vi head_tail.c

```
#include<stdio.h>
#include<stdlib.h>
int main(int ac,char *av[])
{
    if(ac!=2)
    {
        printf("Invalid no. of argument\n");
        return EXIT_FAILURE;
    }
```

```c
FILE *fp1;
if((fp1=fopen("file1","rb"))==NULL)
{
    printf("file name not valid\n");
    return EXIT_FAILURE;
}
char ch;
int line=0;
int n;
while((ch=getc(fp1))!=EOF)
{
    if(ch=='\n')
    {
        n++;
    }
}
rewind(fp1);
int sline=n-5;
printf("\nsline:%d line numbers:%d\n",sline,n);
if(av[1][1]=='h')
{
    while(ch=getc(fp1)!=EOF)
    {
        if(ch=='\n')
        {
            line++;
        }
        if(line<5)
        {
            putc(ch,stdout);
        }
    }
}
rewind(fp1);
if(av[1][1]='t')
{
    line=0;
    while((ch=getc(fp1))!=EOF)
    {
        if(ch=='\n')
        {
            line++;
        }
        if(line>=sline)
```

```
            {
                putc(ch,stdout);
            }
        }
    }
    return EXIT_SUCCESS;
}
```

16.



A string with multiple characters that are repeated consecutively. You're supposed to reduce the size of this string using mathematical logic given as in the example below :

Input :

aabbbbeeeeffggg

Output:

a2b4e4f2g3

Input :

abbcccccc

Output:

ab2c5

vi multi_char.c

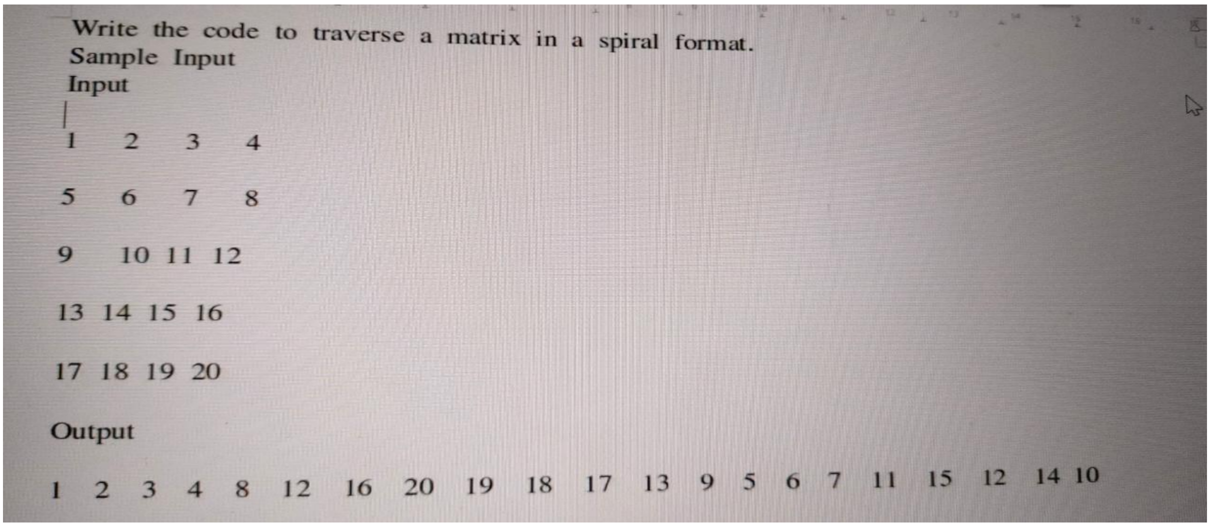
```
#include<stdio.h>
#include<string.h>
void multi_char(char *s)
{
    int c;
    for(int i=0;i<strlen(s);i++)
    {
        c=1;
        for(int j=i+1;s[i]==s[j];j++,c++)
            if(c==1)
                printf("%c",s[i]);
            else
                printf("%c%d",s[i],c);
        i+=c;
    }
    printf("\n");
}
int main()
{
    char str[20];
    printf("Enter a string:");
    scanf("%s",str);
```

```
        multi_char(str);
}
```

Another method

```
#include <stdio.h>
int  show(char *str)
{
        int ctr=1;
        for(char *p=str;(*p);p++)
        {
                if(*p==*(p+1))
                {
                        while(*p==*(p+1))
                        {
                                p++;
                                ctr++;
                        }
                        printf("%c%d",*p,ctr);
                        ctr=1;
                }
                else
                        printf("%c",*p);
        }
}
int main()
{
        char str[20];
        scanf("%s",str);
        show(str);
}
```

17.



Write the code to traverse a matrix in a spiral format.

Sample Input

Input

```
1   2   3   4

5   6   7   8

9   10  11  12

13  14  15  16

17  18  19  20
```

Output

```
1  2  3  4  8  12  16  20  19  18  17  13  9  5  6  7  11  15  12  14  10
```

vi spiral.c

```
#include<stdio.h>
#define r 4
#define c 5
int main()
{
      int a[5][4]={{1,2,3,4},
                   {5,6,7,8},
                   {9,10,11,12},
                   {13,14,15,16},
                    {17,18,19,20}};
       int i,left=0,right=c-1,top=0,bottom=r-1;
       while(left<=right&&top<=bottom)
       {
             for(i=left;i<=right;i++)
             {
                   printf("%d ",a[top][i]);
             }
             top++;
             for(i=top;i<=bottom;i++)
             {
                   printf("%d ",a[i][right]);
             }
             right--;
             if(top<=bottom)
             {
                   for(i=right;i>=left;i--)
                   {
                         printf("%d ",a[bottom][i]);
                   }
                   bottom--;
             }
             if(left<=right)
             {
                   for(i=bottom;i>=top;i--)
                   {
                         printf("%d ",a[i][left]);
                   }
                   left++;
             }
       }
       return 0;
}
```

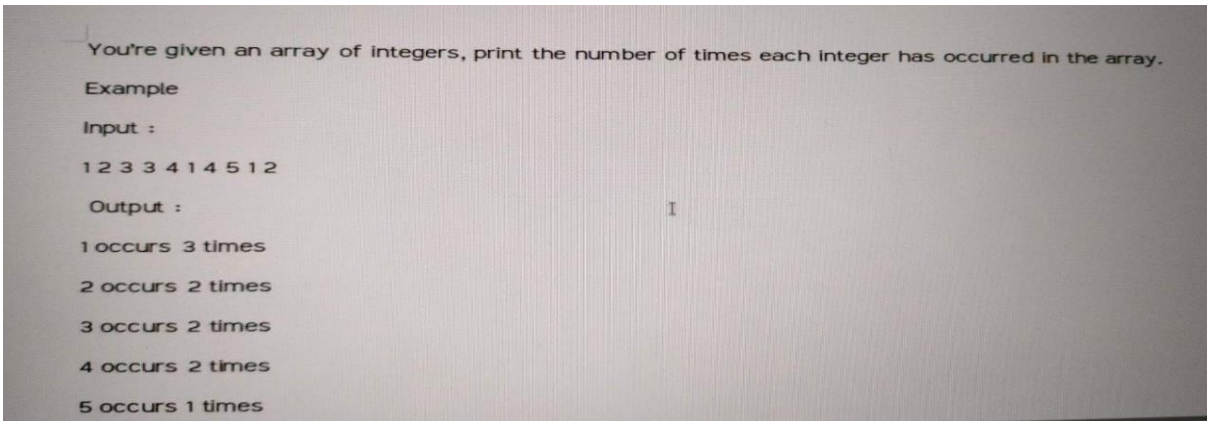
Another method

```
#include<stdio.h>

int main()
```

```
{
   int a[5][4] = {{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16},{17,18,19,20}};
   int rs = 0, re = 5, cs = 0, ce = 4;
   int i, j, k=0;
   for(i=0;i<5;i++)
   {
      for(j=0;j<4;j++)
      {
         printf("%d\t",a[i][j]);
      }
      printf("\n");
   }
  printf("\n");
   while(rs<re && cs<ce)
   {
      for(i=cs;i<ce;i++)
      {
         printf("%3d",a[rs][i]);
      }
      rs++;

      for(i=rs;i<re;i++)
      {
         printf("%3d",a[i][ce-1]);
      }
      ce--;

      if(rs<re)
      {
         for(i=ce-1; i>=cs; --i)
         {
            printf("%3d", a[re - 1][i]);
            }
            re--;
        }
         if(cs<ce)
        {
            for(i=re-1; i>=rs; --i)
            {
                printf("%3d", a[i][cs]);
            }
            cs++;
        }
    }
    return 0;
}
```

```
INPUT: 1  2  3   4
             5   6   7   8
            9  10   1 1  12
           13  14  15   16
            17   18   19    20
OUTPUT: 1  2   3  4  8  12   16   20   19   18   17   13   9  5  6   7   11   15  14  10
```

18.



You're given an array of integers, print the number of times each integer has occurred in the array.

Example

Input :

1 2 3 3 4 1 4 5 1 2

Output :

1 occurs 3 times

2 occurs 2 times

3 occurs 2 times

4 occurs 2 times

5 occurs 1 times

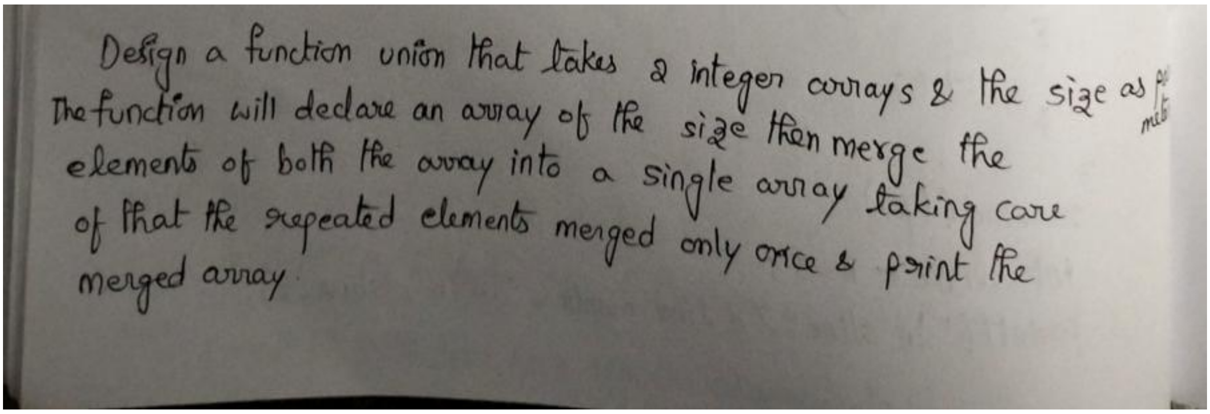
vi arr_times.c

```
#include <stdio.h>
int main()
{
```

```
int n;
scanf("%d",&n);
int arr[n];
int count;
int k=0;
for(int i=0; i<n; i++)
{
    scanf("%d",&arr[i]);        //input of arrray
}
int newarr[n];
//step 1
for(int i=0; i<n; i++)
{
    count = 0;
    for(int j=0; j<=i; j++)
    {
        if(arr[i]==arr[j])
        {
            count++;
        }
    }
    if(count==1)
    {
        newarr[k] = arr[i];
        k++;
    }
}
//step 2
for(int i=0; i<k; i++)
{
    count = 0;
    for(int j=0; j<n; j++)
```

```
        {
            if(newarr[i]==arr[j])
            {
                count++;
            }
        }
        printf("%d occurs %d times\n",newarr[i],count);
    }
    return 0;
}
```

19.



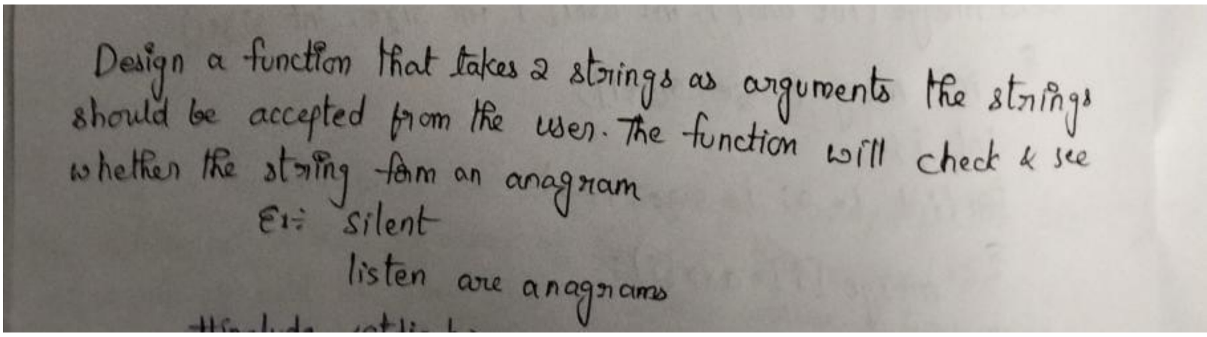
Design a function union that takes 2 integer arrays & the size as parameter
The function will declare an array of the size then merge the
elements of both the array into a single array taking care
of that the repeated elements merged only once & print the
merged array

vi union_arr.c

```
#include<stdio.h>
void print_union(int arr1[],int arr2[],int m,int n)
{
    int i=0,j=0;
    while(i<m&&j<n)
    {
        if(arr1[i]<arr2[j])
            printf("%d ",arr1[i++]);
        else if(arr2[j]<arr1[i])
            printf("%d ",arr2[j++]);
        else
        {
                printf("%d ",arr2[j++]);
                    i++;
            }
    }
    while(i<m)
        printf("%d ",arr1[i++]);
    while(j<n)
        printf("%d ",arr2[j++]);
    printf("\n");
}
int main()
{
    int arr1[]={1,2,4,5,6};
    int arr2[]={2,3,5,7,8};
    int m=sizeof(arr1)/sizeof(arr1[0]);
    int n=sizeof(arr2)/sizeof(arr2[0]);
    print_union(arr1,arr2,m,n);

}
```

OUTPUT: 1   2   3 4   5 6 7 8

20.



Design a function that takes 2 strings as arguments the strings should be accepted from the user. The function will check & see whether the string form an anagram

Ex:- silent
listen are anagrams

vi anagram.c

```
#include<stdio.h>
#include<string.h>
int anagram(char str1[],char str2[]);
int main()
```

```
{
    char str1[50],str2[50];
    int count;
    printf("Enter the first string:");
    scanf("%s",str1);
    printf("Enter the second string:");
    scanf("%s",str2);
    count=anagram(str1,str2);
    if(count==1)
    {
        printf("%s and %s strings are an anagram\n",str1,str2);
    }
    else
    {
        printf("%s and %s strings are not an anagram\n",str1,str2);
    }
    return 0;
}
int anagram(char str1[],char str2[])
{
    int num1[20]={0},num2[20]={0},i=0;
    for(i=0;str1[i]!='\0';i++)
    {
        num1[str1[i]-'a']++;
    }
    for(i=0;str2[i]!='\0';i++)
    {
        num2[str2[i]-'a']++;
    }
    for(i=0;i<20;i++)
    {
        if(num1[i]!=num2[i])
```

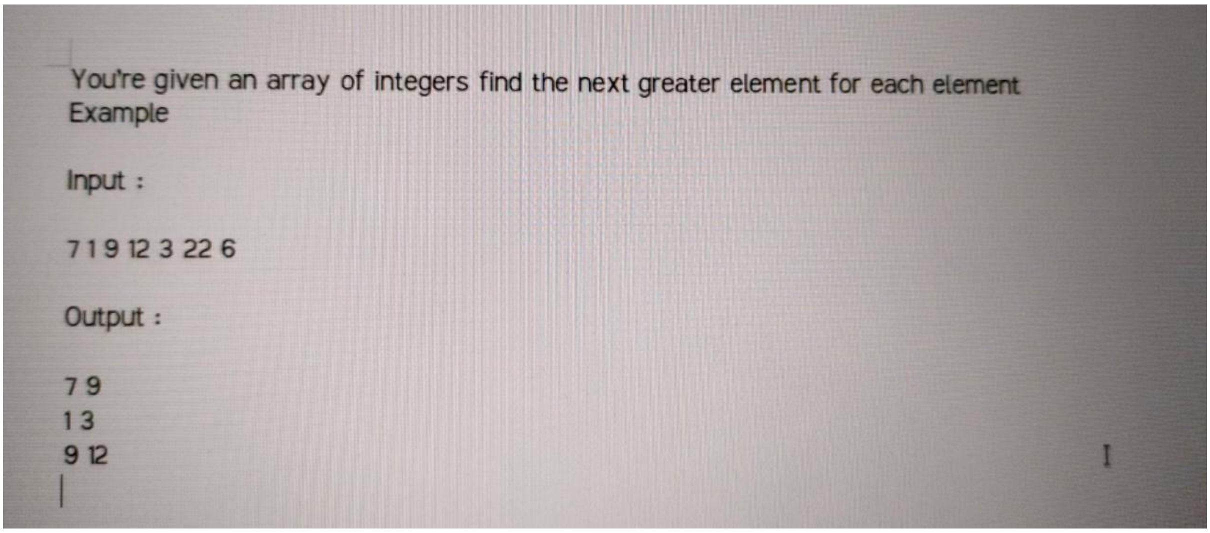
```
                return 0;
        }
        return 1;
}
```

INPUT: Enter first string:listen

Enter second string:silent

OUTPUT: listen and silent are an anagram

21.



You're given an array of integers find the next greater element for each element
Example

Input :

```
7 1 9 12 3 22 6
```

Output :

```
7 9
1 3
9 12
```

vi next_greater.c

```
#include <stdio.h>
int main()
{
        int arr[6]={7,2,11,9,1,22};
        unsigned int gt;
        unsigned int num;
        for(int i=0;i<6;i++)
        {
             gt=10000;
              for(int j=0;j<6;j++)
              {
              //     gt=arr[i]-arr[i+1];
                     if(arr[i]<arr[j] &&(arr[j]-arr[i])<gt )
                     {
```

```
                gt=arr[j]-arr[i];
                num=arr[j];
            }

        }

            printf("\n%3d--%3d",arr[i],num);
            num=0;
    }
}
```

```
OUTPUT: 7----9
          1------3
           9-------12
```