1.Group anagram:

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
void sort(char* a)
{
   int n = strlen(a);
   for (int i = 0; i < n - 1; i++)
   {
      for (int j = i + 1; j < n; j++)
      {
         if (a[i] > a[j])
         {
            char temp = a[i];
            a[i] = a[j];
            a[j] = temp;
         }
      }
   }
}
int isAnagram(char* a, char* b)
{
   char i[100];
   strcpy(i,a);
   char j[100];
   strcpy(j,b);
   int len1 = strlen(i);
   int len2 = strlen(j);
   if (len1 != len2)
      return 0;
   for (int k = 0; k < len1; k++)
   {
      i[k] = tolower(i[k]);
      j[k] = tolower(j[k]);
   }
   sort(i);
   sort(j);
   for (int k = 0; k < len1; k++)
   {
```

```
        if (i[k] != j[k])
            return 0;
    }
    return 1;
}
int main()
{
    char arr[][40] = {"eat","tea","tan","ate","nat","bat"};
    int n = sizeof(arr) / sizeof(arr[0]);
    for(int i=0; i<n; i++){
        printf("%s ",arr[i]);
        for(int j=i+1; j<n-1; j++){
            if(isAnagram(arr[i],arr[j])){
                printf("%s ",arr[j]);
                for(int k=j; k<n-1; k++){
                strcpy(arr[k], arr[k+1]);
                }
                n--;
                j--;
            }
        }
        printf("\n");
    }
    return 0;
}
```

2.Odd even using recursion

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
void even(int a,int b)
{
    if(b>a)
    {
    if(b%2)
    even(a,b-1);
    else
    even(a,b-2);
    if(b%2==0)
```

```
    printf("%d ",b);
    }
}
void odd(int a,int b)
{
    if(b>a)
    {
        if(b%2)
        odd(a,b-2);
        else
        odd(a,b-1);
        if(b%2)
        printf("%d ",b);
    }
}
int main()
{
    int a=2,b=25;
    printf("Even:\n");
    even(a,b);
    printf("\nOdd:\n");
    odd(a,b);
    return 0;
}
```

3.Grid projection:

```
#include <stdio.h>
#include <limits.h>
int nonzero(int m,int n,int a[m][n])
{
    int c=0;
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
        {
            if(a[i][j]!=0)
            c++;
        }
    }
```

```
    return c;
}
int colmax(int m,int n,int a[m][n])
{
    int maximum=INT_MIN,s=0;
    for(int j=0;j<n;j++)
    {
        maximum=INT_MIN;
        for(int i=0;i<m;i++)
        {
            if(a[i][j]>maximum)
            maximum=a[i][j];
        }
        s+=maximum;
    }
    return s;
}
int rowmax(int m,int n,int a[m][n])
{
    int maximum=INT_MIN,s=0;
    for(int i=0;i<m;i++)
    {
        maximum=INT_MIN;
        for(int j=0;j<n;j++)
        {
            if(a[i][j]>maximum)
            maximum=a[i][j];
        }
        s+=maximum;
    }
    return s;
}
int gridprojection(int m,int n,int a[m][n])
{
    int x=nonzero(m,n,a);
    int y=colmax(m,n,a);
    int z=rowmax(m,n,a);
    return x+y+z;
}
int main()
```

```
{
   int m,n;
   scanf("%d %d",&m,&n);
   int a[m][n];
   for(int i=0;i<m;i++)
   {
      for(int j=0;j<n;j++)
      {
         scanf("%d",&a[i][j]);
      }
   }
   printf("%d",gridprojection(m,n,a));
   return 0;
}
```

4.Circular prime:

```
#include <stdio.h>
#include <math.h>
int isPrime(int n)
{
   for(int i=2;i*i<=n;i++)
   {
      if(n%i==0)
      return 0;
   }
   return 1;
}
int circularprime(int n)
{
   int c=0;
   int t=n;
   while(t>0)
   {
      c++;
      t/=10;
   }
   int num=n;
   while(isPrime(num))
   {
```

```
        int r=num%10;
        int d=num/10;
        num=(int)pow(10,c-1)*r+d;
        if(num==n)
        return 1;
    }
    return 0;
}
int main()
{
    int n;
    scanf("%d",&n);
    if(circularprime(n))
    printf("Yes");
    else
    printf("No");
    return 0;
}
```

5.Edit distance:

```
#include <stdio.h>
#include <string.h>
int min(int a,int b,int c)
{
    if(a<=b && a<=c)
    return a;
    if(b<=c && b<=a)
    return b;
    return c;
}
int editDistance(int m,int n,char a[m],char b[n])
{
    int dp[m+1][n+1];
    for(int i=0;i<=m;i++)
    {
        for(int j=0;j<=n;j++)
        {
            if(i==0)
            dp[i][j]=j;
```

```
            else if(j==0)
            dp[i][j]=i;
            else if(a[i-1]==b[j-1])
            dp[i][j]=dp[i-1][j-1];
            else
            dp[i][j]=1+min(dp[i][j-1],dp[i-1][j],dp[i-1][j-1]);
        }
    }
    return dp[m][n];
}
int main()
{
    char a[50],b[50];
    scanf("%[^\n]%*c",&a);
    scanf("%[^\n]%*c",&b);
    int m=strlen(a);
    int n=strlen(b);
    printf("%d",editDistance(m,n,a,b));
    return 0;
}
```

6.Longest palindromic substring:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int palindrome(int s,int e,char c[])
{
    while(s<=e)
    {
        if(c[s]!=c[e])
        return 0;
        s++;
        e--;
    }
    return 1;
}
char* LongestPalindromicSubstring(char* c)
{
    int n=strlen(c),maxlen=1;
```

```
    char* maxstr=&c[0];
    for(int i=0;i<n;i++)
    {
        for(int j=i+maxlen;j<n;j++)
        {
            if(j-i>maxlen && palindrome(i,j-1,c))
            {
                maxlen=j-i;
                maxstr=&c[i];
            }
        }
    }
    char* result=(char*)malloc(maxlen+1);
    strcpy(result,maxstr);
    result[maxlen]='\0';
    return result;
}
int main()
{
    char c[100];
    scanf("%[^\n]%*c",&c);
    printf("%s",LongestPalindromicSubstring(c));
    return 0;
}
```

7.Sum of odd numbers in an array using recursion:

```
#include <stdio.h>
int oddsum(int n,int a[n])
{
     if (n == 0)
        return 0;
    int currentSum = (a [n - 1] % 2 != 0) ? a [n - 1] : 0;
    return currentSum + oddsum( n - 1,a);
}
int main()
{
    int a[]={1,3,2,4};
    int n=4;
    printf("%d",oddsum(n,a));
```

```
    return 0;
}
```

8.Equilibrium index in array:

```
#include <stdio.h>
int sum(int s,int e,int v[])
{
    int sum=0;
    for(int i=s;i<=e;i++)
    {
        sum+=v[i];
    }
    return sum;
}
int ind(int n,int v[n])
{
    for(int i=1;i<n;i++)
    {
        int a=sum(0,i,v);
        int b=sum(i+1,n-1,v);
        if(a==b)
        return i;
    }
}
int main()
{
    int n;
    scanf("%d",&n);
    int v[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&v[i]);
    }
    printf("%d",ind(n,v));
    return 0;
}
```

9.Valid prime anagrams in a range:

```
#include <stdio.h>
int isPrime(int n)
{
   for(int i=2;i*i<=n;i++)
   {
      if(n%i==0)
      return 0;
   }
   return 1;
}
int isAnagram(int a,int b)
{
   int c[10]={0};
   while(a>0)
   {
      c[a%10]++;
      a/=10;
   }
   while(b>0)
   {
      c[b%10]--;
      b/=10;
   }
   for(int i=0;i<10;i++)
   {
      if(c[i]!=0)
      return 0;
   }
   return 1;
}
int main()
{
   int m,n,k=0;
   int p[100];
   scanf("%d %d",&m,&n);
   for(int i=m;i<=n;i++)
   {
      if(isPrime(i))
      p[k++]=i;
   }
```

```
    p[k]='\0';
    for(int i=0;i<k;i++)
    {
        for(int j=i+1;j<n-1;j++)
        {
            if(isAnagram(p[i],p[j]))
            printf("%d %d\n",p[i],p[j]);
        }
    }
    return 0;
}
```

10.Product of maximum three integers in an array:

```
#include <stdio.h>
void sort(int n,int v[n])
{
    for(int i=0;i<n;i++)
    {
        for(int j=i+1;j<n;j++)
        {
            if(v[i]>v[j])
            {
                int temp=v[i];
                v[i]=v[j];
                v[j]=temp;
            }
        }
    }
}
int maxproduct(int n,int v[n])
{
    sort(n,v);
    int b=v[0]*v[1]*v[n-1];
    int a=v[n-1]*v[n-2]*v[n-3];
    if(a>b)
    return a;
    else
    return b;
}
```

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

10.Find all permutations of a given string:

```
#include <stdio.h>
#include <string.h>
void generatePermutation(char *str,const int start, int end)
{
  char temp;
  int i,j;
  for(i = start; i < end-1; ++i){
  for(j = i+1; j < end; ++j)
  {
    temp = str[i];
  str[i] = str[j];
    str[j] = temp;
  generatePermutation(str , i+1 ,end);
    temp = str[i];
    str[i] = str[j];
    str[j] = temp;
  }
  }
  printf("%s\n",str);
}
int main()
{
  char str[] = "ABC";
  int n =strlen(str);
  generatePermutation(str,0,n);
```

```
}
```

10.Deleting the anagrams:

```
#include <stdio.h>
#include<string.h>
void sort(char a[],int n)
{
   for(int i=0;i<n;i++)
   {
      for(int j=i+1;j<n;j++)
      {
         if(a[i]>a[j])
         {
            char temp=a[i];
            a[i]=a[j];
            a[j]=temp;
         }
      }
   }
}
int isAnagram(char a[],char b[])
{
   int len1=strlen(a),len2=strlen(b);
   char temp1[100],temp2[200];
   strcpy(temp1,a);
   strcpy(temp2,b);
   sort(temp1,len1);
   sort(temp2,len2);
   if(len1!=len2)
   return 0;
   for(int i=0;i<len1;i++)
   {
      if(temp1[i]!=temp2[i])
      return 0;
   }
   return 1;
}
int main()
{
```

```
    int n;
    scanf("%d",&n);
    char v[n][100];
    for(int i=0;i<n;i++)
    {
       scanf("%s",&v[i]);
    }
    for(int i=0;i<n;i++)
    {
       printf("%s ",v[i]);
       for(int j=i+1;j<n;j++)
       {
          if(isAnagram(v[i],v[j]))
          {
             for(int k=j;k<n-1;k++)
             {
                strcpy(v[k],v[k+1]);
             }
             j--;
             n--;
          }
       }
    }
    return 0;
}
```