Q1: WAP that will merge two arrays without duplication.

A)

#include<stdio.h>

void merge(int n1,int n2,int a[],int b[])

{

int i,c,k=0,j;

int e[n1+n2];

for (i=0;i<n1;i++)

{

c=0;

for (j=0;j<k;j++)

{

if (e[j]==a[i])

{

c=1;

break;

}

}

if (c!=1)

{

e[k]=a[i];

k++;

}

}

for (i=0;i<n2;i++)

{

c=0;

for (j=0;j<k;j++)

{

if (e[j]==b[i])

{

c=1;

break;

}

}

if (c!=1)

{

e[k]=b[i];

k++;

}

}

for(i=0;i<k;i++)

printf("%d ",e[i]);

}

int main()

{

int n1,n2,i;

scanf("%d",&n1);

scanf("%d",&n2);

int a[n1],b[n2];

for (i=0;i<n1;i++)

{

scanf("%d",&a[i]);

}

for (i=0;i<n2;i++)

{

scanf("%d",&b[i]);

}

merge(n1,n2,a,b);

}

Q2) WAP to print the common elements of two arrays.

A)

#include<stdio.h>

void common(int n1,int n2,int a[],int b[])

{

int i,j,c;

for (i=0;i<n1;i++)

{

c=0;

for (j=0;j<n2;j++)

{

if (a[i]==b[j])

{

c++;

b[j]=-1;

}

}

if (c!=0)

printf("%d ",a[i]);

}

}

int main()

{

int n1,n2,i;

scanf("%d",&n1);

scanf("%d",&n2);

int a[n1],b[n2];

for (i=0;i<n1;i++)

{

scanf("%d",&a[i]);

}

for (i=0;i<n2;i++)

{

scanf("%d",&b[i]);

}

common(n1,n2,a,b);

}

Q3) You are provided with an array of n responses (whose values range from 1 to9) for a survey. Write a program to draw a histogram of the survey response results.

A)

#include<stdio.h>

void frequency(int n,int a[])

{

int i,j,c;

for (i=1;i<=9;i++)

{

c=0;

for (j=0;j<n;j++)

{

if (a[j]==i)

c++;

}

printf("%d frequency = %d\n",i,c);

}

}

int main()

{

int i,n;

scanf("%d",&n);

int a[n];

for (i=0;i<n;i++)

scanf("%d",&a[i]);

frequency(n,a);

}

Q4) WAP that contains a function which will print the number given as argument in words – 123 to be printed as ONE TWO THREE.

A)

#include<stdio.h>

void words(int n)

{

int x;

while (n!=0)

{

x=x\*10+(n%10);

n=n/10;

}

if (x==0)

printf("Zero");

else

{

while(x!=0)

{

if (x%10==0)

printf("ZERO ");

else if(x%10==1)

printf("ONE ");

else if(x%10==2)

printf("TWO ");

else if(x%10==3)

printf("THREE ");

else if(x%10==4)

printf("FOUR ");

else if(x%10==5)

printf("FIVE");

else if(x%10==6)

printf("SIX ");

else if(x%10==7)

printf("SEVEN ");

else if(x%10==8)

printf("EIGHT ");

else if(x%10==9)

printf("NINE ");

x=x/10;

}

}

}

int main()

{

int n;

scanf("%d",&n);

words(n);

}

Q5) WAP with a function that accepts an array and the number of elements in the array as inputs and returns the sum of the elements in the array.

A)

#include<stdio.h>

int sum(int arr[],int n)

{

int x=0,i;

for (i=0;i<n;i++)

x+=arr[i];

return x;

}

int main()

{

int n,s,i;

scanf("%d",&n);

int a[n];

for (i=0;i<n;i++)

scanf("%d",&a[i]);

s=sum(a,n);

printf("%d",s);

}

Q6) WAP that contains a function that accepts an array and the number of elements in the array as inputs and returns the difference between largest and smallest elements of the array.

A)

#include<stdio.h>

int diff(int a[],int n)

{

int i,j,sub,t,min;

for (i=0;i<n;i++)

{

min=i;

for(j=i+1;j<n;j++)

{

if (a[j]<a[min])

{

min=j;

}

}

t=a[min];

a[min]=a[i];

a[i]=t;

}

sub=a[n-1]-a[0];

return sub;

}

int main()

{

int i,n,d;

scanf("%d",&n);

int a[n];

for (i=0;i<n;i++)

scanf("%d",&a[i]);

d=diff(a,n);

printf("%d",d);

}

Q7) Given a string S, check whether it’s a Pangram or not. A pangram is a string that contains all alphabets at least once.

A)

#include<stdio.h>

#include<string.h>

int panagram(char s[50])

{

int i,flag,j;

for (i=97;i<123;i++)

{

flag=0;

for (j=0;j<strlen(s);j++)

{

if (s[j]==i)

{

flag=1;

}

}

if (flag==0)

{

flag=2;

return 0;

break;

}

}

if (flag!=2)

return 1;

}

int main()

{

char s1[50];

int v;

gets(s1);

v=panagram(s1);

if (v==0)

{

printf("Not panagram");

}

else

{

printf("Panagram");

}

}

Q8) Given two strings, check whether they are anagrams or not. Anagrams are words with same alphabets with different ordering.

A)

#include<stdio.h>

#include<string.h>

int anagram(char s1[50],char s2[50])

{

int len,i,j,min;

char t;

len=strlen(s2);

if (strlen(s1)==strlen(s2))

{

for (i=0;i<len;i++)

{

min=i;

for (j=i+1;j<len;j++)

{

if (s1[j]<s1[min])

{

min=j;

}

}

t=s1[i];

s1[i]=s1[min];

s1[min]=t;

//s[i]=s[min];

}

s1[i]='\0';

for (i=0;i<len;i++)

{

min=i;

for (j=i+1;j<len;j++)

{

if (s2[j]<s2[min])

{

min=j;

}

}

t=s2[i];

s2[i]=s2[min];

s2[min]=t;

}

//puts(s1);

//puts(s2);

if (strcmp(s1,s2)==0)

return 0;

else

return 1;

}

else

return 0;

}

int main()

{

char s[50],r[50];

int v,len;

gets(s);

//puts(s);

gets(r);

//puts(r);

len=strlen(r);

s[len]='\0';

v=anagram(s,r);

if (v==0)

printf("Anagram");

else

printf("Not Anagram");

}

Q9) Write a program to remove all characters that are not alphabets from a given string.

A)

#include<stdio.h>

#include<string.h>

void alpha(char s[50])

{

int i,len,j=0;

char r[50];

len=strlen(s);

for (i=0;i<len;i++)

{

if (s[i]>='a'&&s[i]<='z'||s[i]>='A'&&s[i]<='Z')

{

r[j]=s[i];

j++;

}

}

puts(r);

}

int main()

{

char s[50],r[50];

int v,len;

gets(s);

alpha(s);

}

Q10) Given a string S, write a program to find the repeated character present first in S.

A)

#include<stdio.h>

#include<string.h>

void repeat(char s[50])

{

int i,len,j,flag=0;

char r[50];

len=strlen(s);

for (i=0;i<len;i++)

{

for (j=i+1;j<len;j++)

{

if (s[i]==s[j])

flag=1;

}

if (flag==1)

break;

}

if (flag==1)

printf("%c",s[i]);

else

printf("no repeated character");

}

int main()

{

char s[50];

gets(s);

repeat(s);

}