SET – 4 ARRAYS

Q10)WAP that will merge two arrays without duplication.

#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]);

}

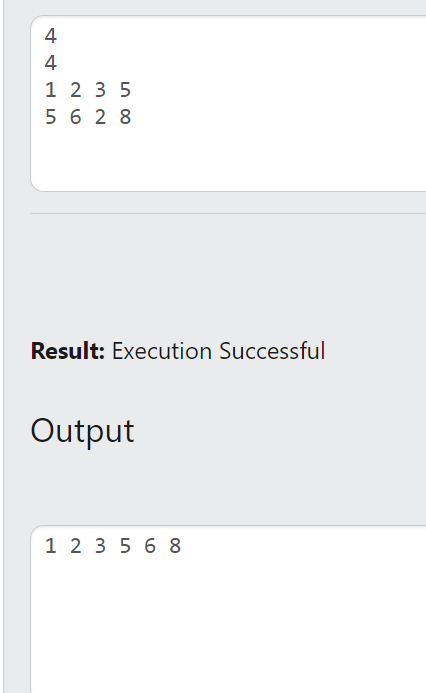
for (i=0;i<n2;i++)

{

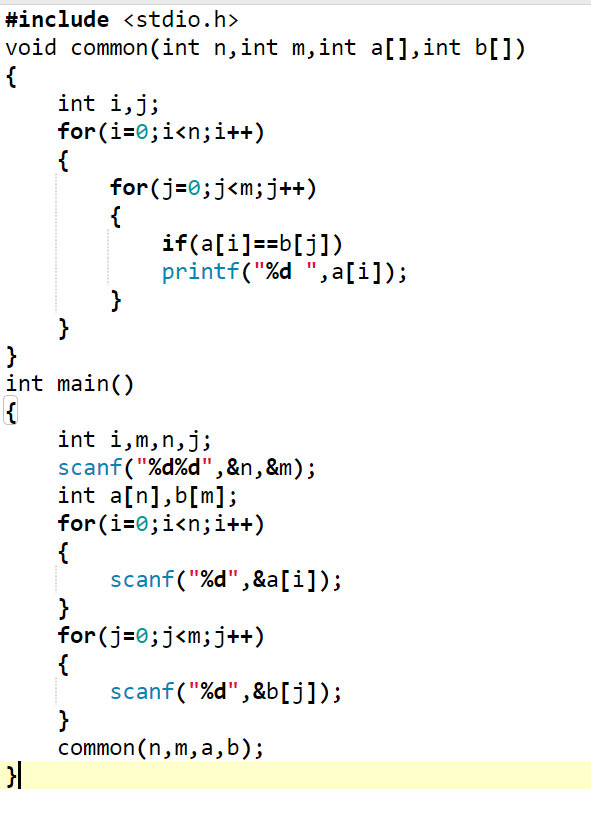
scanf("%d",&b[i]);

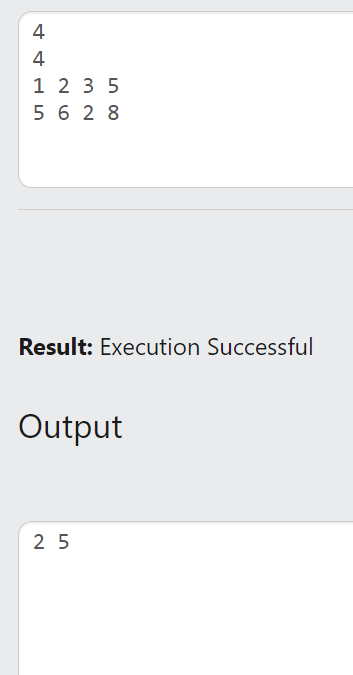
}

merge(n1,n2,a,b);



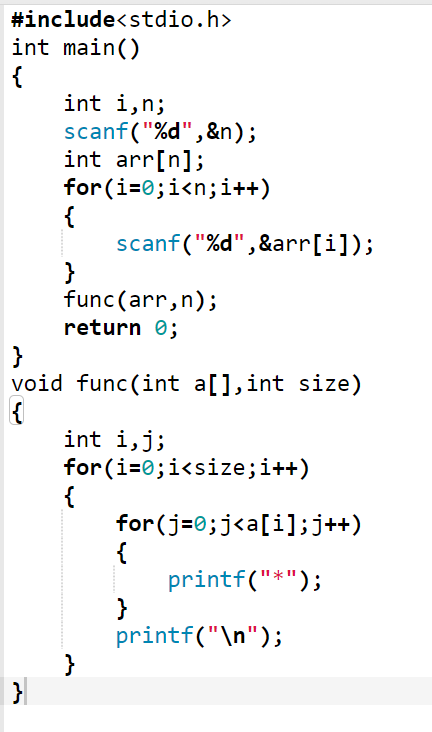
Q11) WAP to print the common elements of two arrays.

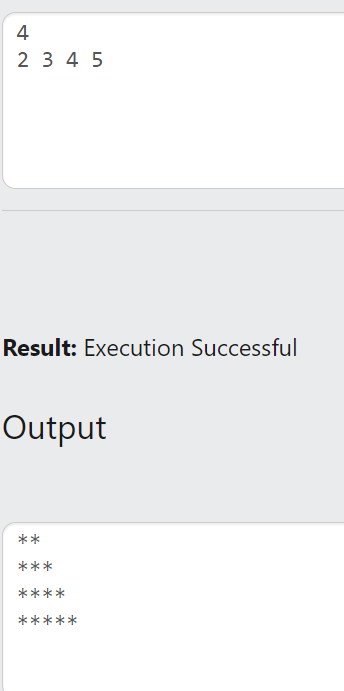




Q12) You are provided with an array of n responses (whose values range from 1 to9) for a

survey. Write a program to draw a histogram of the survey response results.





Q13) Given an array of nonnegative integers, write a program to find the minimum number of

elements such that their sum should be greater than the sum of the rest of the elements of

the array. Given {3, 1, 7, 1} the output should be 1 element (i.e. {7}) since 7 is greater

than the sum of the rest of the elements i.e 3+1+1 = 5.

#include <stdio.h>

int findMinElements(int arr[], int n) {

int totalSum = 0;

int currentSum = 0;

int count = 0;

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

totalSum += arr[i];

}

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (arr[j] < arr[j + 1]) {

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

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

currentSum += arr[i];

count++;

if (currentSum > totalSum - currentSum) {

break;

}

}

return count;

}

int main() {

int n,i;

printf("enter the n value\n");

scanf("%d",&n);

printf("enter array elements\n");

int arr[n];

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

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

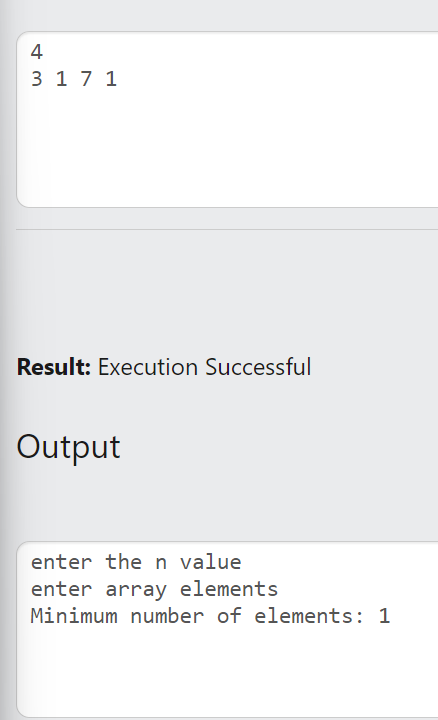
}

int minElements = findMinElements(arr, n);

printf("Minimum number of elements: %d\n", minElements);

return 0;

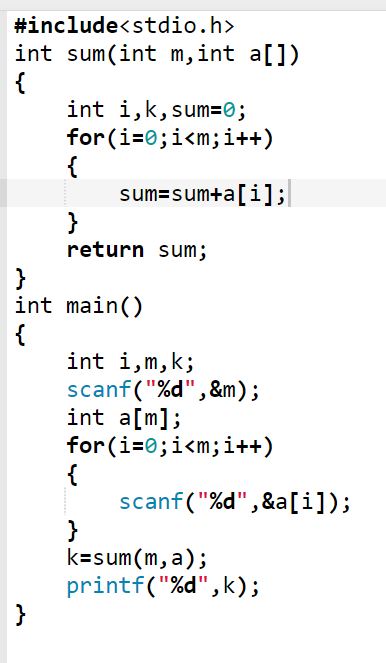
}

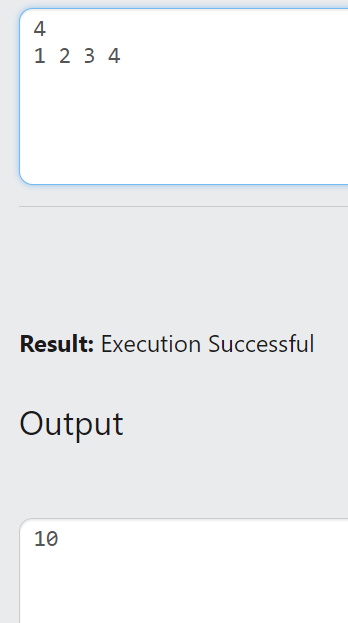


SET – 5 SIMPLE FUNCTIONS

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





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

#include <stdio.h>

int findDifference(int arr[], int n) {

if (n <= 0) {

printf("Error: Array is empty.\n");

return 0;

}

int smallest = arr[0];

int largest = arr[0];

for (int i = 1; i < n; i++) {

if (arr[i] < smallest) {

smallest = arr[i];

}

if (arr[i] > largest) {

largest = arr[i];

}

}

return largest - smallest;

}

int main() {

int n,i;

printf("enter the n value\n");

scanf("%d",&n);

int arr[n];

printf("enter the array elements\n");

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

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

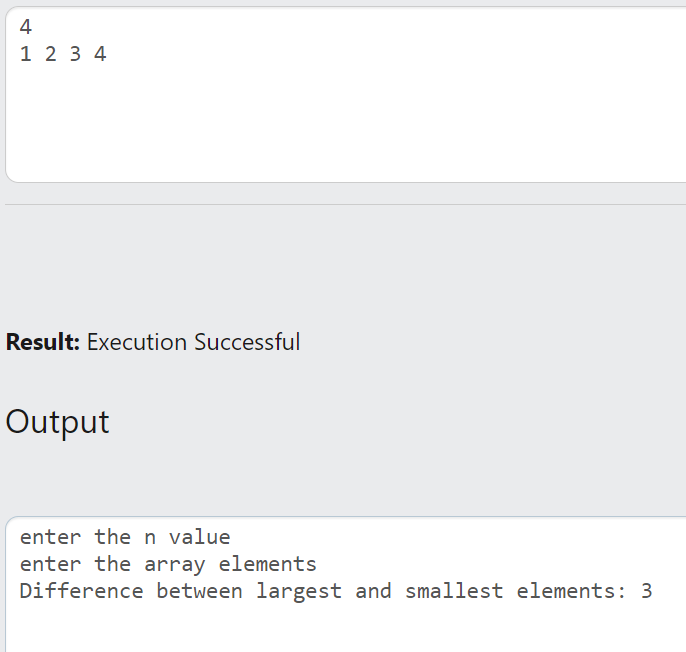
}

int difference = findDifference(arr,n);

printf("Difference between largest and smallest elements: %d\n", difference);

return 0;

}



SET -8 STRINGS

Q5) Given a string S, check whether it’s a Pangram or not. A pangram is a string that contains

all alphabets at least once

#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");

}

}

Q6) Given two strings, check whether they are anagrams or not. Anagrams are words with

same alphabets with different ordering. (Examples: (dog, god), (leg, gel), (ant, tan),( dam,

mad)).

#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");

}

Q7) Write a program to remove all characters that are not alphabets from a given string

#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);

}

Q8) Given a string S, write a program to find the repeated character present first in S (Not

the first repeated character!!).

Input: geeksforgeeks

output: g

#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);

}