Tribhuvan University

Institute of Engineering

Thapathali Campus,Thapathali

LAB SHEET #4

Submitted by:

Name: Kishan Adhikari

Roll No. :THA077BCT021

Submitted to:

Department of Electronics and Computer Engineering

Date : 7th August 2021

Title:

Write a program to find separately the sum of the positive and negative integer elements of an array of size 10. Pass this array to a function called sort_array(int[]) and display the array elements into ascending order.

Problem Analysis:

The problem is to find the sum of positive and negative numbers in an array of size 10 and sort the array into ascending order using different functions. So ,To accomplish the task we define a user defined function as:

```
void sort_array(int n[10]);
```

After that we define integer *i* and set *positive_sum* and *negative_sum* variables to zero.We use *for* loop and *scanf* function to store value inside array and calculate *positive_sum* and *negative_sum* by using logic as:

```
if n[i] is positive( >0)
Add to positive_sum
else
Add to negative_sum
```

Calculating the positive_sum and negative_sum part is accomplished. Now the next task is to sort the array into an ascending order. To sort arrays in ascending order ,different sorting algorithms such as Bubble sort, Quick sort, Binary sort, Insertion sort etc are already there. We are sorting using the Bubble sort algorithm. The logic is as:

```
Bubble_sort(arr[10]):
        for(i from to 9)
                for (j form 0 to 8)
                        if (num[j]>num[j+1])
                        Swap (num[j],num[j+1])
```

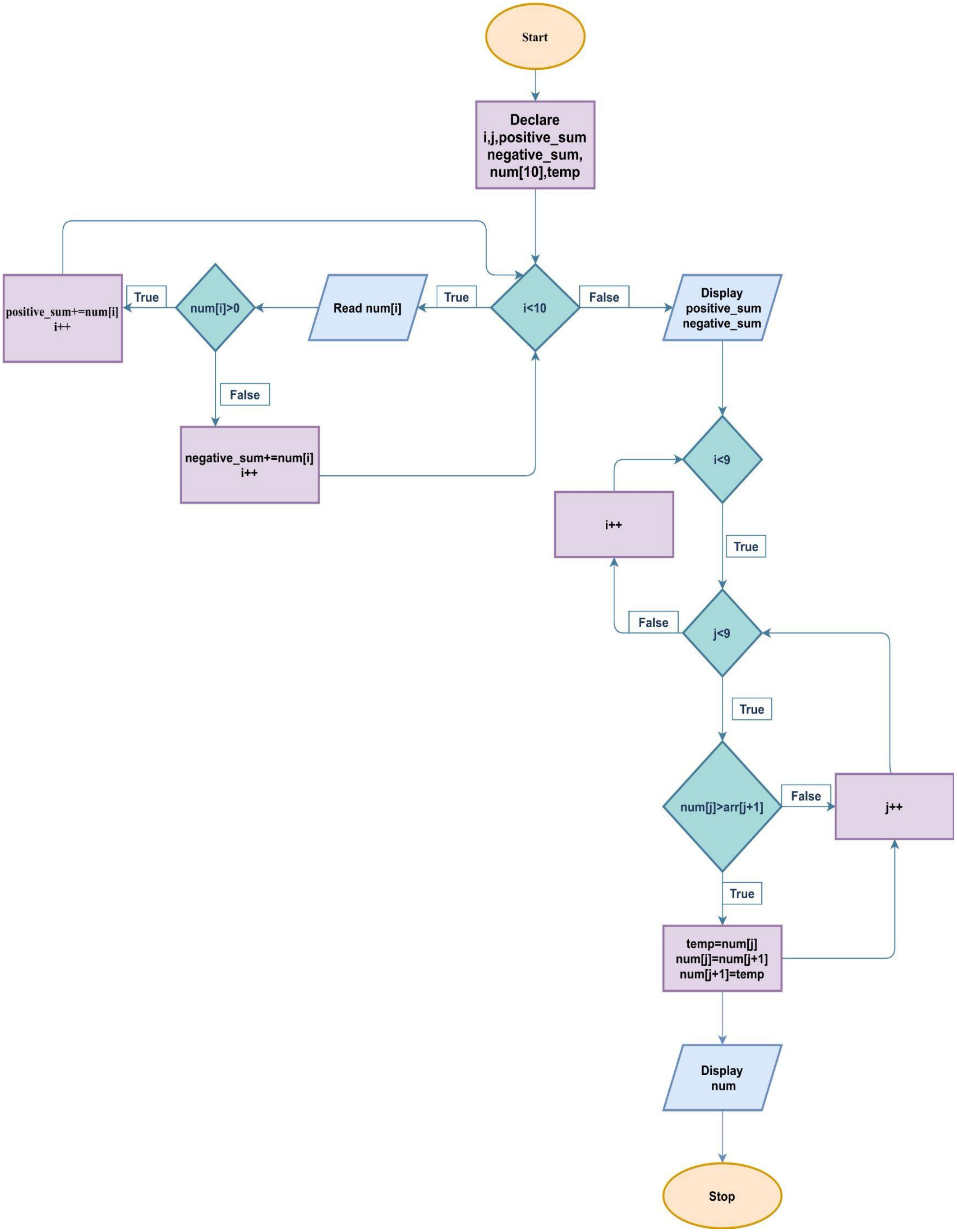
Input variables	Function used	Output variables	Header files

i,j(int)
arr[10] (int)
	printf()
	scanf()
	sort_array(int arr[10])
		arr[i]
		positive_sum(int)
		Negative_sum(int)
			stdio.h

Algorithm:

```
1.  Start
2.  Declare function as: sort_array(int n[10])
3.  Define variables of int as: i,j,temp,arr[10] positive_sum,negative_sum
4.  for (i from 0 to 9)
5.  If (n[i] >0) : positive_sum+=n[i] else goto 6
6.  negative_sum[i]+=n[i]
7.  Display positive sum,negative sum
8.  Goto 9
9.  for (i from 0 to 9)
10. for (j from 0 to 8)
11. If (n[j]>n[j+1]) true goto 12
12. temp = arr[j];
13. arr[j] = arr[j + 1];
14. arr[j + 1] = temp;
15. Display n
16. stop
```

Flowchart:

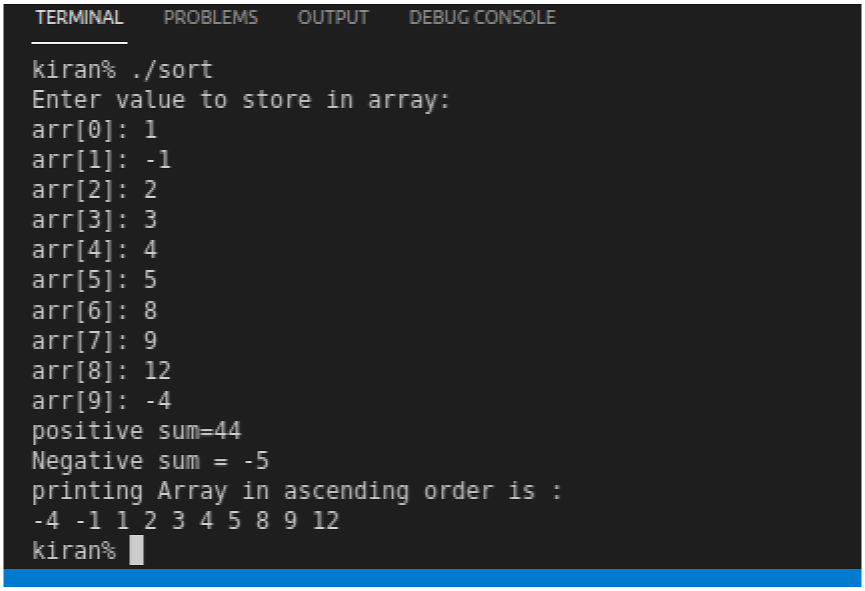

Start
Declare i,j,positive_sum negative_sum, num[10],temp
i<10
True
Read num[i]
num[i]>0
True
positive_sum+=num[i]
i++
False
negative_sum+=num[i]
i++
False
Display positive_sum negative_sum
i<9
i++
True
j<9
False
True
num[j]>arr[j+1]
False
j++
True
temp=num[j]
num[j]=num[j+1]
num[j+1]=temp
Display num
Stop

Source Code:

```
/* @Author: Kishan Adhikari
   @Created Date: 2078/05/04
   @filename: sortarray.c
  @Description:program to find separately the sum of the positive and `
negative integer elements of an array of size 10. Pass this array to a
function called sort_array(int[])and display the array elements into
ascending order.
 */
#include <stdio.h>
void sort_array(int arr[10]);
int main()
{
 int i, positive_sum = 0, negative_sum = 0;
 int arr[10];

 printf("Enter value to store in array:\n");
 for (i = 0; i < 10; i++)
 {
   printf("arr[%d]: ", i);
   scanf("%d", arr + i);
   if (arr[i] > 0)
   {
     positive_sum += arr[i];
   }
   if (arr[i] < 0)
   {
     negative_sum += arr[i];
   }
 }
 printf("positive sum=%d\n", positive_sum);
 printf("Negative sum = %d\n", negative_sum);
 sort_array(arr);
 printf("printing Array in ascending order is :\n");
 for (i = 0; i < 10; i++)
 {
   printf("%d ", arr[i]);
 }
 printf("\n");
}
```

```
void sort_array(int arr[10])
{
 int i, j, temp = 0;

 for (i = 0; i < 9; i++)
 {

   for (j = 0; j < 9; j++)
   {
     if (arr[j] > arr[j + 1])
     {
       temp = arr[j];
       arr[j] = arr[j + 1];
       arr[j + 1] = temp;
     }
   }
 }
}
```

Output:



```
kiran% ./sort
Enter value to store in array:
arr[0]: 1
arr[1]: -1
arr[2]: 2
arr[3]: 3
arr[4]: 4
arr[5]: 5
arr[6]: 8
arr[7]: 9
arr[8]: 12
arr[9]: -4
positive sum=44
Negative sum = -5
printing Array in ascending order is :
-4 -1 1 2 3 4 5 8 9 12
kiran% 
```

Discussion and Conclusion:

From this lab ,I understood the basics of array , calculation using array and different sorting algorithms to sort an array of integer type in ascending as well as descending order.

Title:

Write a program to find separately the sum of the positive and negative integer elements of an array of size 10. Pass the positive and negative elements to separate functions e.g.:sum_positive(int*), sum_negative(int*) to carry out its sum. Also pass this array to a function called sort_array(int[]) and display the array elements into ascending order using pointer.

Problem Analysis:

The problem is to find separately the sum of positive and negative integer as well as sort element of an array of size 10 using pointer and different function for different task. To solve this problem we first define three function with argument as:

```
void sum_positive(int *arr, int n);
void sum_negative(int *arr, int n);
void sort_array(int *arr, int n);
```

We store value inside *arr pointer, declare variable as sum,i and calculate sum_positive and sum_negative using logic as:

*sum_positive(int *arr, int n);*

```
for(i=0 to n)
if (*(arr+i)>0) then
sum+= *(arr+i)
```

*sum_negative(int *arr, int n);*

```
for(i=0 to n)
if (*(arr+i)<0) then
sum+= *(arr+i)
```

Then after that we sort the array using the Bubble sort algorithm. The logic is as:

```
for (i = 0; i < n; i++)
   for (j = i + 1; j < n; j++)
     if (*(arr + j) < *(arr + i))
       temp = *(arr + i);
       *(arr + i) = *(arr + j);
       *(arr + j) = temp;
```

Then we display array elements using a loop.

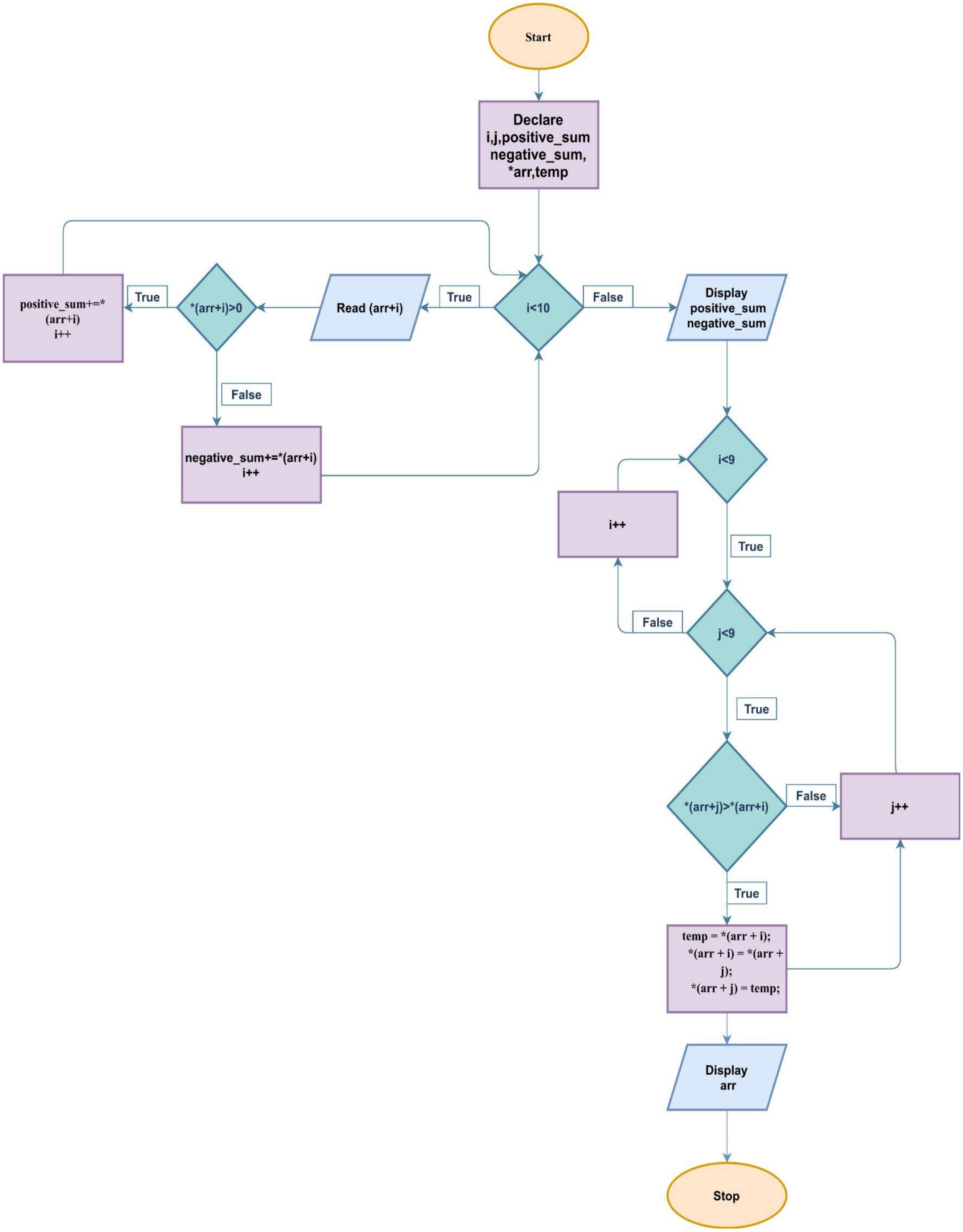
Input variables	Processing variables/ calculations	Output variables	Necessary header files and function

*arr(int)
temp(int)
sum1(int)
sum2(int)
I(int)
j(int)
	sum1+=*(arr+i)
	sum2+=*(arr+i)
		sum1(int)
		sum2(int)
		*arr(int)
			stdio.h
			sum_positive(int*arr,int n)
			sum_negative(int*arr,int n)
			sort_array(int*arr,int n)

Algorithm:

```
1.  Start
2.  Declare variable as : *arr,i,j,temp,n,sum1,sum2
3.  Read n
4.  for(i=0 to n) read (arr+i) goto 5
5.  If (arr[i] is positive) sum1+=*(arr+i) goto 6
6.  if(arr[i] is negative) sum2+=*(arr+i) goto 7
7.  If i=n goto 8 else goto 5
8.  Display sum1,sum2
9.  for (i=0 to n) true goto 10 false goto 16
10. for(j=i+1 to n) true goto 11
11. if  (*(arr+j)<*(arr+i)) true goto 12 false goto 15
12. temp=*(arr+i)
13. *(arr+i)=*(arr+j)
14. *(arr+j)=temp goto 15
15. j++ goto 10
16. i++
17. Display arr
18. Stop
```

Flowchart:

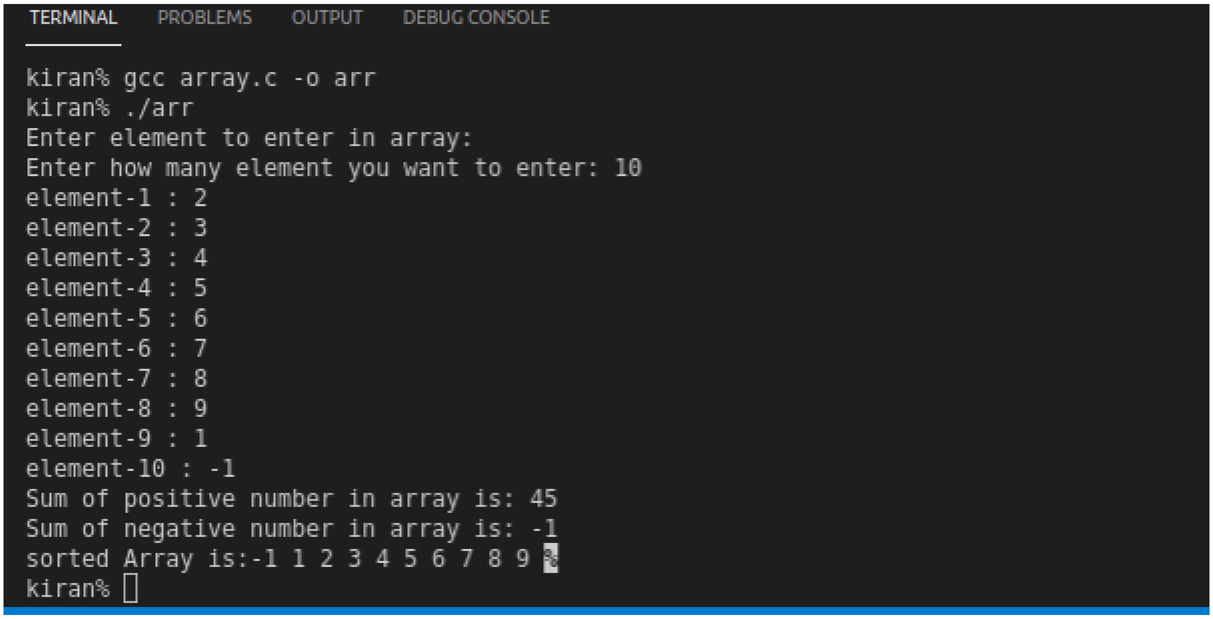

Start
Declare i,j,positive_sum negative_sum, *arr,temp
i<10
True
False
Read (arr+i)
*(arr+i)>0
True
positive_sum+=* (arr+i) i++
False
negative_sum+=*(arr+i) i++
Display positive_sum negative_sum
i<9
i++
True
j<9
False
True
(arr+j)>(arr+i)
False
j++
True
temp = *(arr + i);
*(arr + i) = *(arr + j);
*(arr + j) = temp;
Display arr
Stop

Source Code:

```
/* @Author: Kishan Adhikari
   @Created Date: 2078/05/04
   @filename: array.c
    @Description:program to find separately the sum of the positive and
negative integer elements
    of an array of size 10. Pass the positive and negative elements to
separate functions e.g.:
   sum_positive(int*), sum_negative(int*) to carry out its sum. Also pass
this array to a
    function called sort_array(int[]) and display the array elements into
ascending order using
   pointer
 */
#include <stdio.h>
#include <string.h>
void sum_positive(int *arr, int n);
void sum_negative(int *arr, int n);
void sort_array(int *arr, int n);

int main()
{
 int *a;
 int i, n;
 printf("Enter element to enter in array:\n");
 printf("Enter how many element you want to enter: ");
 scanf("%d", &n);
 for (i = 0; i < n; i++)
 {
   printf("element-%d : ", i + 1);
   scanf("%d", a + i);
 }
 sum_positive(a, n);
 sum_negative(a, n);
 sort_array(a, n);
 printf("sorted Array is:");
 for (i = 0; i < n; i++)
 {
   printf("%d ", *(a + i));
 }
 return 0;
}
void sum_positive(int *arr, int n)
{
 int i, sum = 0;
 for (i = 0; i < n; i++)
 {
   if (*(arr + i) > 0)
   {
     sum += *(arr + i);
   }
 }
 printf("Sum of positive number in array is: %d\n", sum);
}
void sum_negative(int *arr, int n)
{
 int i, sum = 0;
 for (i = 0; i < 10; i++)
 {
   if (*(arr + i) < 0)
   {
     sum += *(arr + i);
   }
 }
 printf("Sum of negative number in array is: %d\n", sum);
}
void sort_array(int *arr, int n)
{
 int i, j, temp;
 for (i = 0; i < n; i++)
 {
   for (j = i + 1; j < n; j++)
   {
     if (*(arr + j) < *(arr + i))
     {
       temp = *(arr + i);
```

```
           *(arr + i) = *(arr + j);
           *(arr + j) = temp;
         }
      }
   }
 }
```

Output:



```
kiran% gcc array.c -o arr
kiran% ./arr
Enter element to enter in array:
Enter how many element you want to enter: 10
element-1 : 2
element-2 : 3
element-3 : 4
element-4 : 5
element-5 : 6
element-6 : 7
element-7 : 8
element-8 : 9
element-9 : 1
element-10 : -1
Sum of positive number in array is: 45
Sum of negative number in array is: -1
sorted Array is:-1 1 2 3 4 5 6 7 8 9 %
kiran% 
```

Discussion and Conclusion:

From this lab ,I understood the basics of array, using pointer as function argument ,calculation using array and different sorting algorithms to sort an array of integer type in ascending as well as descending order.

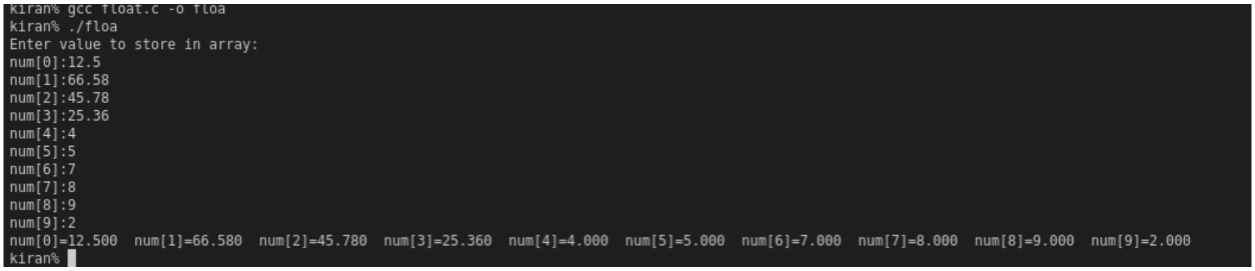
Title:

Write a program to enter 10 floating numbers in an array and display it.

Source code:

```
/*
   @Author: Kishan Adhikari
   @Created Date: 2078/05/05
   @filename: float.c
   @Description:program to enter 10 floating numbers in an array and
display it.
 */
#include <stdio.h>
int main()
{
 int i = 0;
 float num[10];
 printf("Enter value to store in array:\n");
 for (i = 0; i < 10; i++)
 {
   printf("num[%d]:", i);
   scanf("%f", &num[i]);
 }

 for (i = 0; i < 10; i++)
 {
   printf("num[%d]=%.3f  ", i, num[i]);
 }
 printf("\n");
 return 0;
}
```

Output:



```
kiran% gcc float.c -o floa
kiran% ./floa
Enter value to store in array:
num[0]:12.5
num[1]:66.58
num[2]:45.78
num[3]:25.36
num[4]:4
num[5]:5
num[6]:7
num[7]:8
num[8]:9
num[9]:2
num[0]=12.500  num[1]=66.580  num[2]=45.780  num[3]=25.360  num[4]=4.000  num[5]=5.000  num[6]=7.000  num[7]=8.000  num[8]=9.000  num[9]=2.000
kiran% 
```

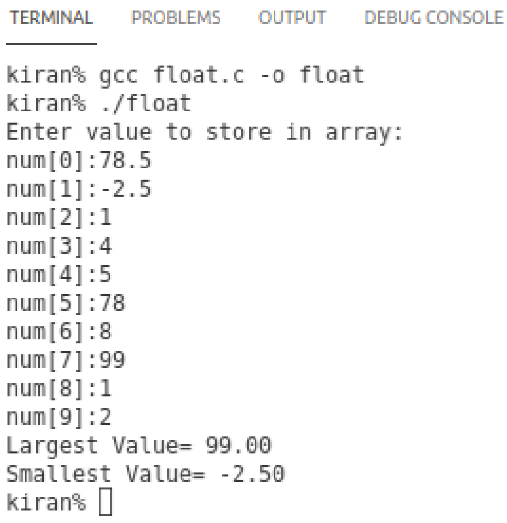
Title:

Write a program to display largest and smallest element of an array defined in Q.No. 1

Source Code:

```
/*
   @Author: Kishan Adhikari
   @Created Date: 2078/05/05
   @filename: float.c
   @Description:program to display largest and smallest element of an
array defined in Q.No. 1
 */
#include <stdio.h>
int main()
{
 int i = 0;
 float num[10], large, small;
 printf("Enter value to store in array:\n");
 for (i = 0; i < 10; i++)
 {
   printf("num[%d]:", i);
   scanf("%f", &num[i]);
 }
 large = num[0];
 small = num[0];
 for (i = 0; i < 10; i++)
 {
   if (num[i] > large)
   {
     large = num[i];
   }
   if (num[i] < small)
   {
     small = num[i];
   }
 }
 printf("Largest Value= %.2f\n", large);
 printf("Smallest Value= %.2f\n", small);

 return 0;
}
```

Output:



```
kiran% gcc float.c -o float
kiran% ./float
Enter value to store in array:
num[0]:78.5
num[1]:-2.5
num[2]:1
num[3]:4
num[4]:5
num[5]:78
num[6]:8
num[7]:99
num[8]:1
num[9]:2
Largest Value= 99.00
Smallest Value= -2.50
kiran% 
```

Title:

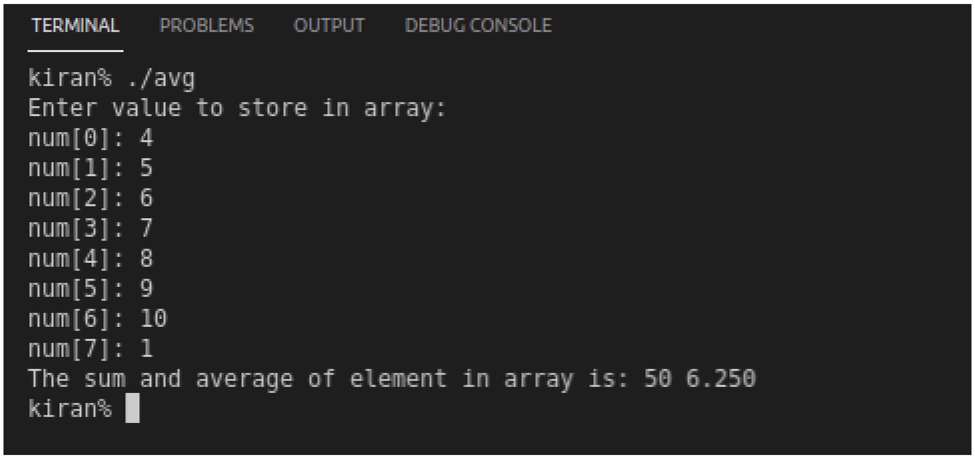
Write a program to initialize a one dimensional array of size 8 and display the sum and average of array elements.

Source Code:

```
/*
   @Author: Kishan Adhikari
   @Created Date: 2078/05/04
   @filename: avgsum.c
   @Description:program to initialize a one dimensional array of size 8
and display
    the sum and average of array elements
```

```
 */
#include <stdio.h>
int main()
{
 int num[8], sum = 0;
 float average = 0;
 //adding data to array
 printf("Enter value to store in array:\n");
 for (int i = 0; i < 8; i++)
 {
   printf("num[%d]: ", i);
   scanf("%d", num + i);
 }
 for (int i = 0; i < 8; i++)
 {
   sum += num[i];
 }
 average = sum / 8.0;
 printf("The sum and average of element in array is: %d %.3f\n", sum,
average);
 return 0;
}
```

Output:



```
kiran% ./avg
Enter value to store in array:
num[0]: 4
num[1]: 5
num[2]: 6
num[3]: 7
num[4]: 8
num[5]: 9
num[6]: 10
num[7]: 1
The sum and average of element in array is: 50 6.250
kiran% 
```

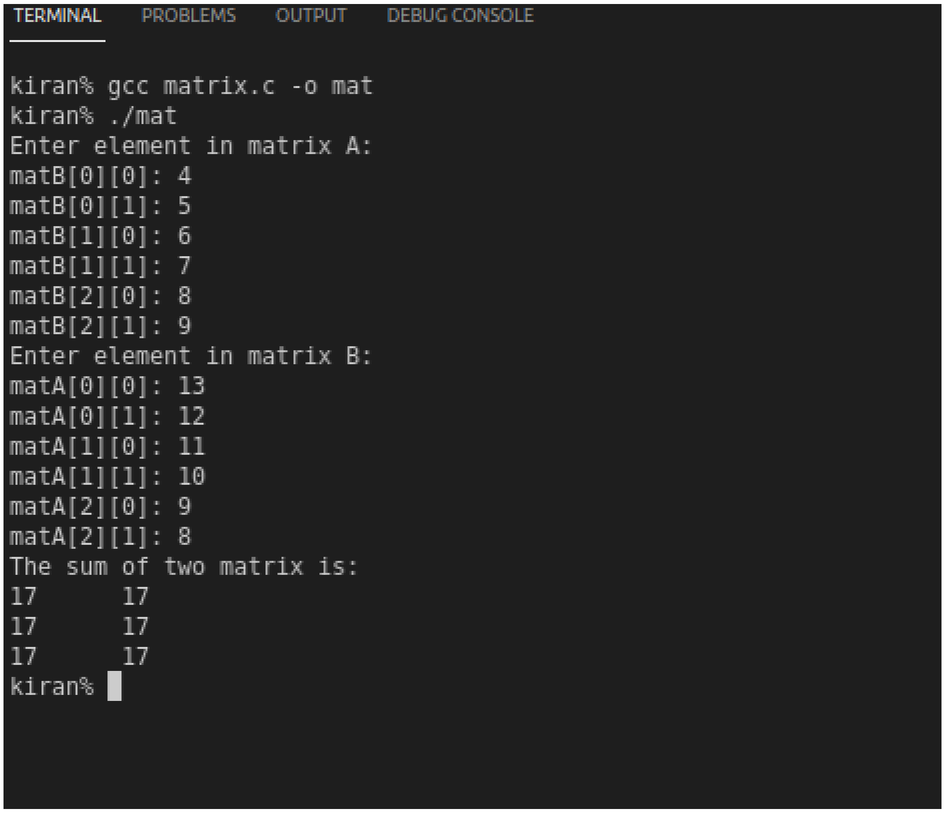
Title:

Write a program to read two matrices of order 3 * 2, add them and display the resultant matrix in matrix form.

Source Code:

```c
/*
   @Author: Kishan Adhikari
   @Created Date: 2078/05/05
   @filename: matrix.c
   @Description:program to read two matrices of order 3 * 2,
    add them and display the resultant matrix in matrix form.

 */

#include <stdio.h>
void add_matrix(int mat1[3][2], int mat2[3][2]);
int main()
{
 int mat1[3][2], mat2[3][2], i, j;
 printf("Enter element in matrix A:\n");
 for (i = 0; i < 3; i++)
 {
   for (j = 0; j < 2; j++)
   {
     printf("matB[%d][%d]: ", i, j);
     scanf("%d", &mat1[i][j]);
   }
 }
 printf("Enter element in matrix B:\n");
 for (i = 0; i < 3; i++)
 {
   for (j = 0; j < 2; j++)
   {
     printf("matA[%d][%d]: ", i, j);
     scanf("%d", &mat2[i][j]);
   }
 }
 add_matrix(mat1, mat2);
}
void add_matrix(int mat1[3][2], int mat2[3][2])
{
 int i, j, add[3][2];
 for (i = 0; i < 3; i++)
 {
   for (j = 0; j < 2; j++)
   {
     add[i][j] = mat1[i][j] + mat2[i][j];
   }
 }
 printf("The sum of two matrix is:\n");
 for (i = 0; i < 3; i++)
 {
   for (j = 0; j < 2; j++)
   {
     printf("%d\t", add[i][j]);
   }
   printf("\n");
 }
}
```

Output:



```
kiran% gcc matrix.c -o mat
kiran% ./mat
Enter element in matrix A:
matB[0][0]: 4
matB[0][1]: 5
matB[1][0]: 6
matB[1][1]: 7
matB[2][0]: 8
matB[2][1]: 9
Enter element in matrix B:
matA[0][0]: 13
matA[0][1]: 12
matA[1][0]: 11
matA[1][1]: 10
matA[2][0]: 9
matA[2][1]: 8
The sum of two matrix is:
17      17
17      17
17      17
kiran% 
```

Title:

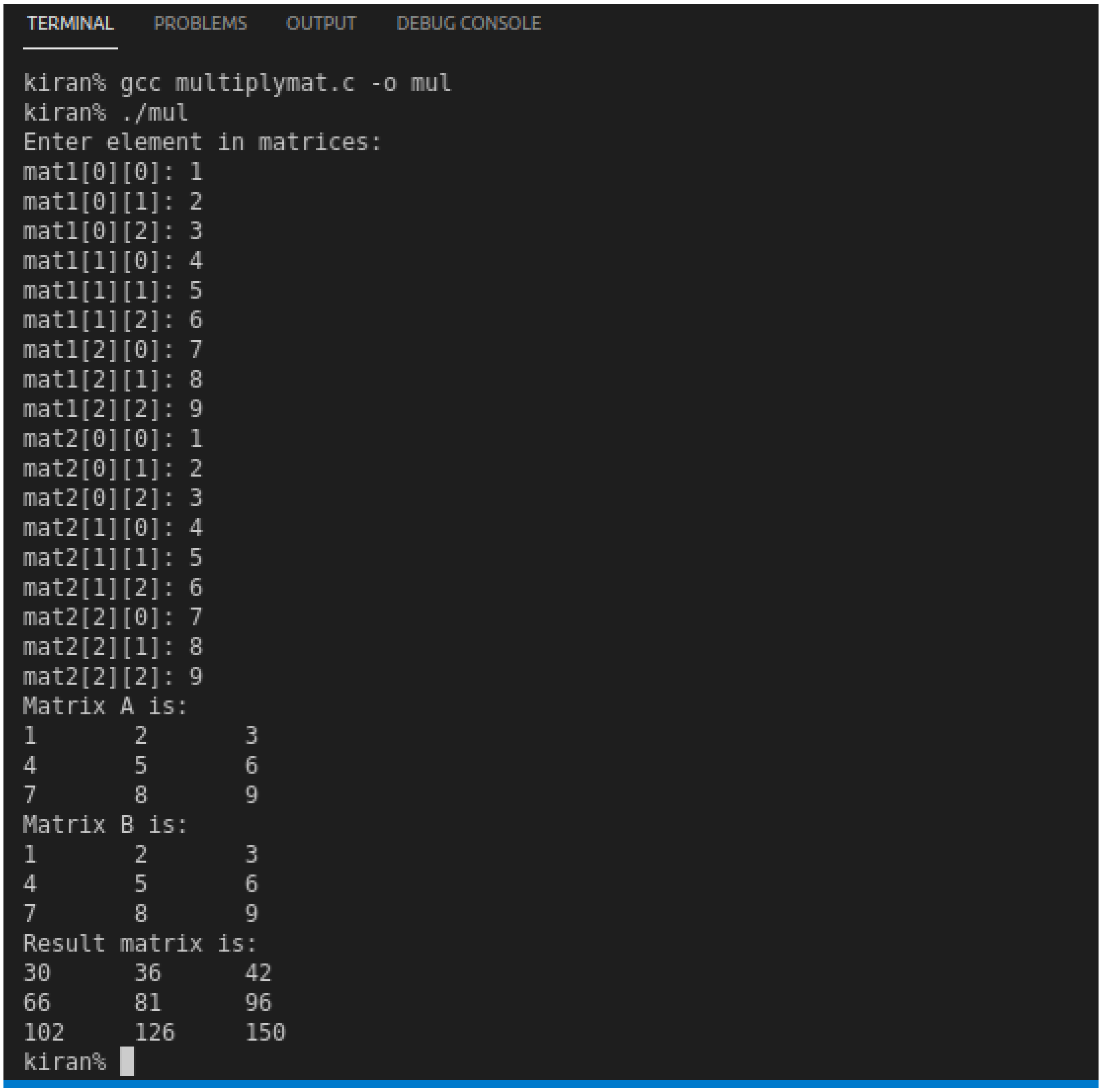
Write a program to multiply two 3*3 matrices.

Source Code:

```c
/*
   @Author: Kishan Adhikari
   @Created Date: 2078/05/06
   @filename: multiplymat.c
   @Description:program to multiply two 3*3 matrices
 */
#include <stdio.h>
int main()
{
 int i, j, k;
 int mat1[3][3];
 int mat2[3][3], result[3][3], sum = 0;
```

```
printf("Enter element in matrices:\n");

for (i = 0; i < 3; i++)
{
  for (j = 0; j < 3; j++)
  {
    printf("mat1[%d][%d]: ", i, j);
    scanf("%d", &mat1[i][j]);
  }
}
for (i = 0; i < 3; i++)
{
  for (j = 0; j < 3; j++)
  {
    printf("mat2[%d][%d]: ", i, j);
    scanf("%d", &mat2[i][j]);
  }
}
printf("Matrix A is:\n");
for (i = 0; i < 3; i++)
{
  for (j = 0; j < 3; j++)
  {
    printf("%d\t", mat1[i][j]);
  }
  printf("\n");
}
printf("Matrix B is:\n");
for (i = 0; i < 3; i++)
{
  for (j = 0; j < 3; j++)
  {
    printf("%d\t", mat2[i][j]);
  }
  printf("\n");
}

//multiplying matrices
for (i = 0; i < 3; i++)
{
   for (j = 0; j < 3; j++)
   {
     sum = 0;
     for (k = 0; k < 3; k++)
     {
       sum += mat1[i][k] * mat2[k][j];
       result[i][j] = sum;
     }
   }
 }
 printf("Result matrix is:\n");
 for (i = 0; i < 3; i++)
 {
   for (j = 0; j < 3; j++)
   {
     printf("%d\t", result[i][j]);
   }
   printf("\n");
 }
 return 0;
}
```

Output:



```
kiran% gcc multiplymat.c -o mul
kiran% ./mul
Enter element in matrices:
mat1[0][0]: 1
mat1[0][1]: 2
mat1[0][2]: 3
mat1[1][0]: 4
mat1[1][1]: 5
mat1[1][2]: 6
mat1[2][0]: 7
mat1[2][1]: 8
mat1[2][2]: 9
mat2[0][0]: 1
mat2[0][1]: 2
mat2[0][2]: 3
mat2[1][0]: 4
mat2[1][1]: 5
mat2[1][2]: 6
mat2[2][0]: 7
mat2[2][1]: 8
mat2[2][2]: 9
Matrix A is:
1       2       3
4       5       6
7       8       9
Matrix B is:
1       2       3
4       5       6
7       8       9
Result matrix is:
30      36      42
66      81      96
102     126     150
kiran% 
```

Title:

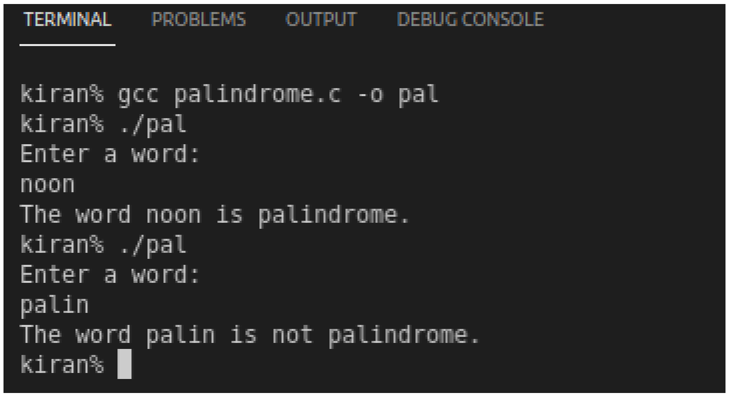
Write a program to read a string and check for palindrome without using string related function (a string is palindrome if its half is mirror by itself eg: abcdcba).

Source Code:

```
/*
   @Author: Kishan Adhikari
   @Created Date: 2078/05/06
   @filename: palindrome.c
   @Description:program to read a string and check for palindrome without
using string related
   function (a string is palindrome if its half is mirror by itself eg:
abcdcba).

 */

#include <stdio.h>

int stringlen(char s[])
{
 int i;
 int len = 0;
 for (i = 0; s[i] != '\0'; i++)
 {
   len++;
 }

 return len;
}
int main()
{

 char word[120];
 int i, end, middle;
 printf("Enter a word:\n");

 scanf("%s", word);
 int size = stringlen(word);
 end = size - 1;
 middle = size / 2;
 // printf("size=%d\n", size);
 for (i = 0; i < middle; i++)
 {
   if (word[i] != word[end])
```

```
    {
      printf("The word %s is not palindrome.\n", word);
      break;
    }
    end--;
  }

  if (i == middle)
  {
    printf("The word %s is palindrome.\n", word);
  }

  return 0;
}
```

Output:



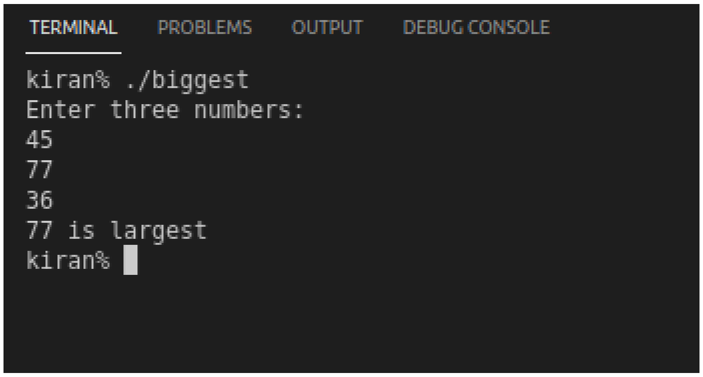
```
kiran% gcc palindrome.c -o pal
kiran% ./pal
Enter a word:
noon
The word noon is palindrome.
kiran% ./pal
Enter a word:
palin
The word palin is not palindrome.
kiran% █
```

Title:
Write a program to find the biggest among three numbers using pointer.

Source Code:

```
/*
   @Author: Kishan Adhikari
   @Created Date: 2078/05/06
   @filename: biggest.c
   @Description:Program to find biggest among three numbers using pointer.
 */
#include <stdio.h>
int main()
{
 int n1, n2, n3;
 int *p1, *p2, *p3;

 printf("Enter three numbers:\n");
 scanf("%d%d%d", &n1, &n2, &n3);
 p1 = &n1, p2 = &n2, p3 = &n3;
 if (*p1 > *p2 && *p1 > *p3)
 {
   printf("%d is largest\n", *p1);
 }
 else if (*p2 > *p1 && *p2 > *p3)
 {
   printf("%d is largest\n", *p2);
 }
 else

 {
   printf("%d is largest\n", *p3);
 }
 return 0;
}
```

Output:



```
kiran% ./biggest
Enter three numbers:
45
77
36
77 is largest
kiran% █
```

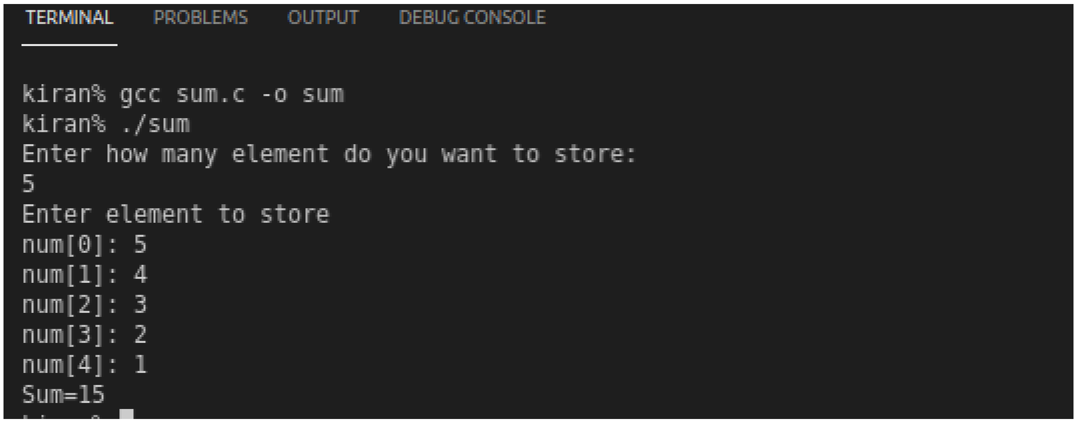
Title:

Write a program to find the sum of all the elements of an array using pointers.

Source Code:

```c
/*
   @Author: Kishan Adhikari
   @Created Date: 2078/05/06
   @filename:sum.c
   @Description:Program to find the sum of all the elements of an array
using pointers.
 */

#include <stdio.h>
#include <stdlib.h>
int main()
{
 int n;
 printf("Enter how many element do you want to store:\n");
 scanf("%d", &n);
 int *num = (int *)malloc(n * sizeof(int));
 int i, sum = 0;
 printf("Enter element to store\n");
 for (i = 0; i < n; i++)
 {
   printf("num[%d]: ", i);
   scanf("%d", num + i);
37
   sum += *(num + i);
 }
 printf("Sum=%d\n", sum);
 free(num);
 return 0;
}
```

Output:



```
kiran% gcc sum.c -o sum
kiran% ./sum
Enter how many element do you want to store:
5
Enter element to store
num[0]: 5
num[1]: 4
num[2]: 3
num[3]: 2
num[4]: 1
Sum=15
```

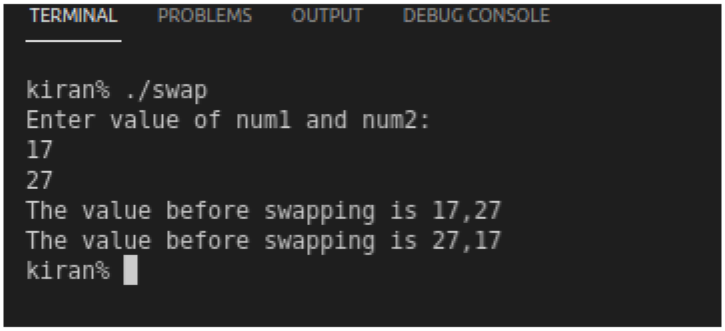
Title:

Write a program to swap value of two variables using pointer.

Source Code:

```
/*
   @Author: Kishan Adhikari
   @Created Date: 2078/05/04
   @filename: swap.c
   @Description:Program to swap value of two variables using pointer.
 */
#include <stdio.h>
int main()
{
 int num1, num2;
 int *p1, *p2, *temp;
 printf("Enter value of num1 and num2:\n");
 scanf("%d%d", &num1, &num2);
 p1 = &num1;
 p2 = &num2;
 printf("The value before swapping is %d,%d\n", num1, num2);
 *temp = *p1;
 *p1 = *p2;
 *p2 = *temp;
 printf("The value before swapping is %d,%d\n", num1, num2);
 return 0;
}
```

Output:



```
kiran% ./swap
Enter value of num1 and num2:
17
27
The value before swapping is 17,27
The value before swapping is 27,17
kiran%
```

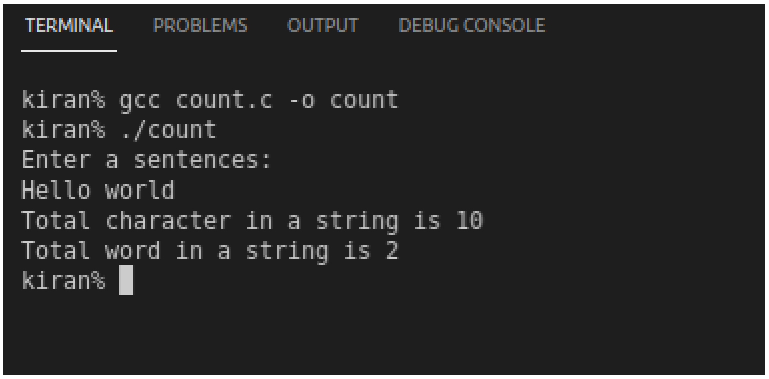
Title:

Write a program to read a sentence and count the number of characters and words in that sentence.

Source Code:

```
/*
   @Author: Kishan Adhikari
   @Created Date: 2078/05/04
   @filename: count.c
   @Description:Program to read a sentence and count the number of
characters
   and words in that sentence.
 */

#include <stdio.h>
#include <string.h>
int main()
{
 char sentences[1024];
 printf("Enter a sentences:\n");
 fgets(sentences, 1024, stdin);
 int word = 0, character = 0, i;
 for (i = 0; i < strlen(sentences); i++)
 {
   if (sentences[i] != ' ')
   {
     character++;
   }
 }
 for (i = 0; i < strlen(sentences); i++)
 {
   if (sentences[i] == 32)
   {
     word++;
   }
 }
 word += 1; //last word doesn't hit space after it .To count last word.
 printf("Total character in a string is %d\n", character - 1);
 printf("Total word in a string is %d\n", word);
}
```

Output:



```
kiran% gcc count.c -o count
kiran% ./count
Enter a sentences:
Hello world
Total character in a string is 10
Total word in a string is 2
kiran% 
```

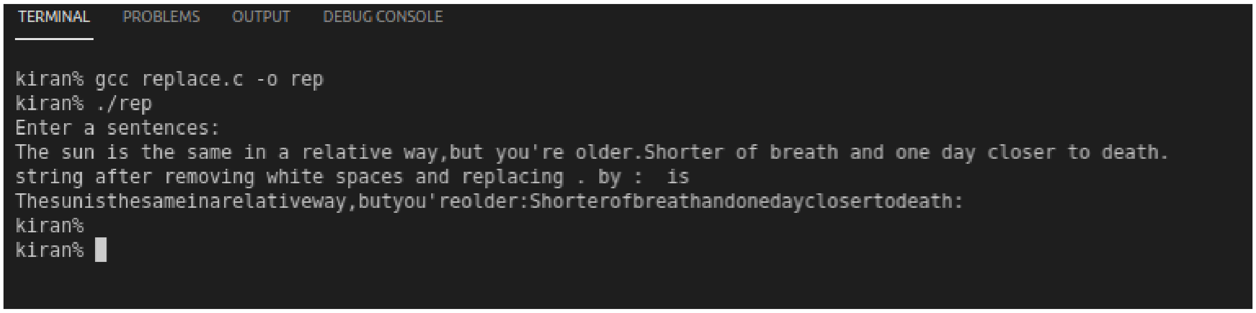
Title:

Write a program to read a sentence and delete all the white spaces. Replace all “.” by “:”.

Source Code:

```
/*  @Author: Kishan Adhikari
   @Created Date: 2078/05/05
   @filename: replace.c
   @Description: program to read a sentence and delete all the white
spaces.
    Replace all “.” by “:”.

 */
#include <stdio.h>
#include <string.h>
int main()
{
 int i, j = 0;
 char sentences[1000];
 printf("Enter a sentences:\n");
 fgets(sentences, 1000, stdin);
 for (i = 0; i < strlen(sentences); i++)
 {
   sentences[i] = sentences[i + j];
   if (sentences[i] == 32)
   {
     j++;
     i--;
   }
   if (sentences[i] == '.')
   {
     sentences[i] = ':';
   }
 }
 printf("string after removing white spaces and replacing . by :  is\n%s",
sentences);
 return 0;
}
```

Output:



```
kiran% gcc replace.c -o rep
kiran% ./rep
Enter a sentences:
The sun is the same in a relative way,but you're older.Shorter of breath and one day closer to death.
string after removing white spaces and replacing . by :  is
Thesunisthesameinarelativeway,butyou'reolder:Shorterofbreathandonedayclosertodeath:
kiran%
kiran% 
```

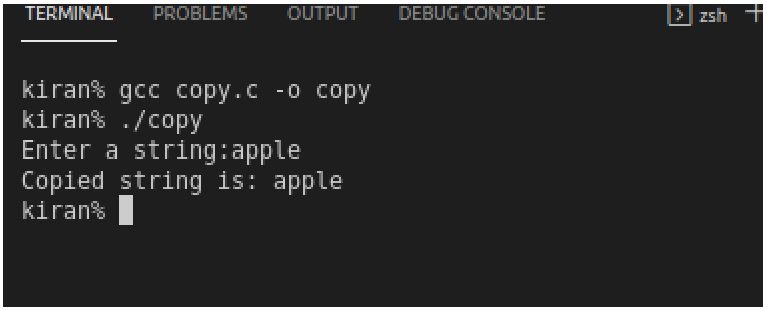
Title:

Write a program to copy one string to another string with and without using string handling function.

Source Code:

```
/*
   @Author: Kishan Adhikari
   @Created Date: 2078/05/04
   @filename: copy.c
   @Description:Program to copy one string to another string with and
without using string handling
   function.
*/
#include <stdio.h>
int main()
{
 char org[1000], copied[1000];
 int i;
 printf("Enter a string:");
 fgets(org, sizeof(org), stdin);
 for (i = 0; org[i] != '\0'; i++)
 {
   copied[i] = org[i];
 }
 copied[i] = '\0';
 printf("Copied string is: %s", copied);
 return 0;
}
```

Output:



```
kiran% gcc copy.c -o copy
kiran% ./copy
Enter a string:apple
Copied string is: apple
kiran%
```

Title:

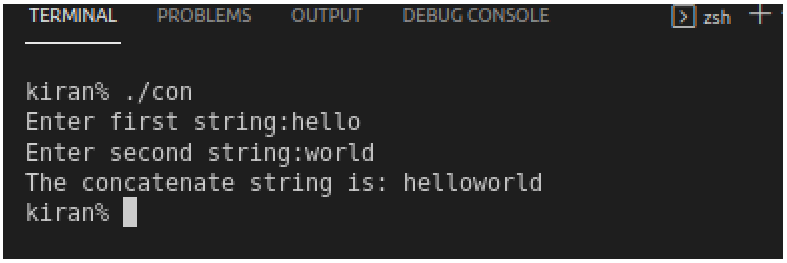
Write a program to concatenate two strings.

Source Code:

```
/*
   @Author: Kishan Adhikari
   @Created Date: 2078/05/06
   @filename: concate.c
   @Description:Program to concatenate two strings.
 */

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

int main()
{
 char str1[100], str2[100];
 printf("Enter first string:");
 // fgets(str1, 100, stdin);
 scanf("%s", str1);
 printf("Enter second string:\n");
 // fgets(str2, 100, stdin);
 scanf("%s", str2);
 strcat(str1, str2); //concatente str2 to str1
```

```c
 printf("The concatenate string is: %s\n", str1);

 return 0;
}
```

Output:



```
kiran% ./con
Enter first string:hello
Enter second string:world
The concatenate string is: helloworld
kiran% 
```

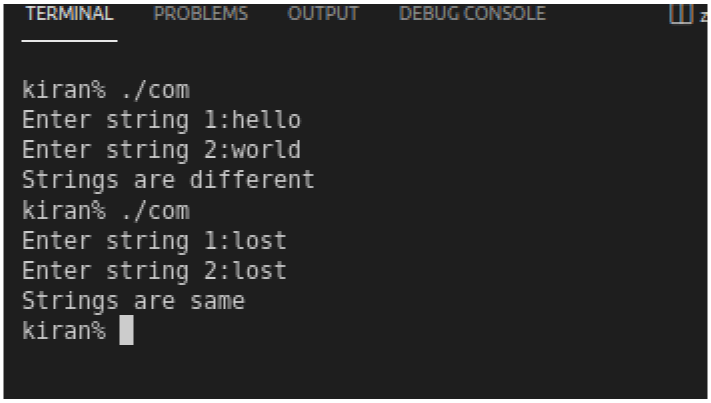
Title:

Write a program to compare two strings.

Source Code:

```c
/*
   @Author: Kishan Adhikari
   @Created Date: 2078/05/04
   @filename: compare.c
   @Description:Program to compare two strings.
 */

#include <stdio.h>
#include <string.h>
int main()
{
 char str1[50], str2[50];
 int value;
 printf("Enter string 1:");
 scanf("%s", str1);

 printf("Enter string 2:");
 scanf("%s", str2);
 value = strcmp(str1, str2);
 if (value == 0)
 {
   printf("Strings are same\n");
 }
 else
 {
   printf("Strings are different\n");
 }
 return 0;
}
```

Output:



```
kiran% ./com
Enter string 1:hello
Enter string 2:world
Strings are different
kiran% ./com
Enter string 1:lost
Enter string 2:lost
Strings are same
kiran% 
```

Title:

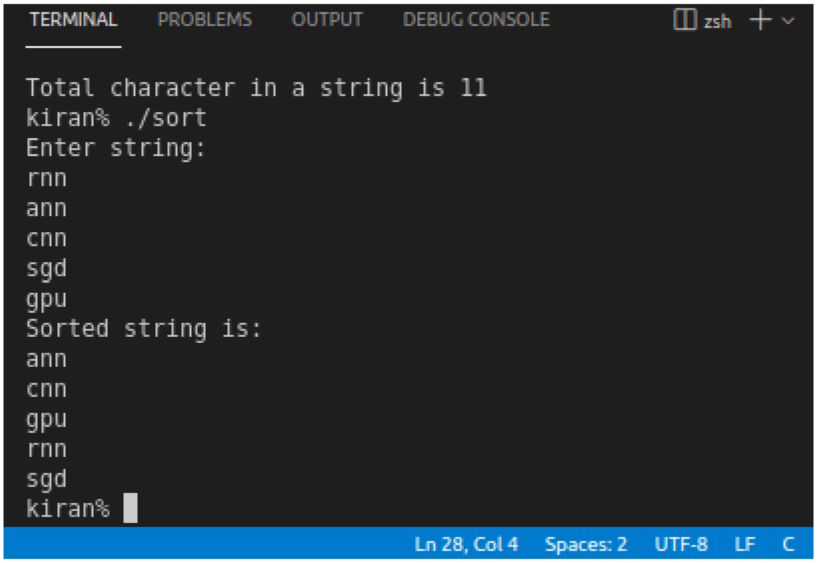
Write a program to sort 5 string words stored in an array of pointers.

Source Code:

```
/* @Author: Kishan Adhikari
  @Filename:sort.c
  @Created Date: 2078/05/07
  @Description: Write a program to sort 5 string words stored in an array
of pointers.
*/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
```

```
void sort(char *x[]);
int main()
{
 char *arr[5], new[5];
 int check;

 printf("Enter string:\n");
 for (int i = 0; i < 5; i++)
 {
   arr[i] = (char *)malloc(100 * sizeof(char));
   scanf("%s", arr[i]);
 }
 for (int i = 0; i < 4; i++)
 {
   for (int j = i + 1; j < 5; j++)
   {
     check = strcmp(arr[i], arr[j]);
     if (check > 0)
     {
       strcpy(new, arr[j]);
       strcpy(arr[j], arr[i]);
       strcpy(arr[i], new);
     }
   }
 }
 printf("Sorted string is:\n");
 for (int i = 0; i < 5; i++)
 {
   printf("%s\n", arr[i]);
 }
 return 0;
}
```

Output:



```
Total character in a string is 11
kiran% ./sort
Enter string:
rnn
ann
cnn
sgd
gpu
Sorted string is:
ann
cnn
gpu
rnn
sgd
kiran%
```