Data Structure Lab Report

Only for course Teacher						
		Needs Improvement	Developing	Sufficient	Above Average	Total Mark
Allocate mark & Percentage		25%	50%	75%	100%	25
Understanding	3					
Analysis	4					
Implementation	8					
Report Writing	10					
					Total obtained mark	
Comments						

Semester: Fall 2024

Student Name: Md. Jakaria Nur

Student ID: 232-35-252

Batch: SWE-41 **Section:** A2

Course Code: SE 132
Course Name: Data Structure Lab

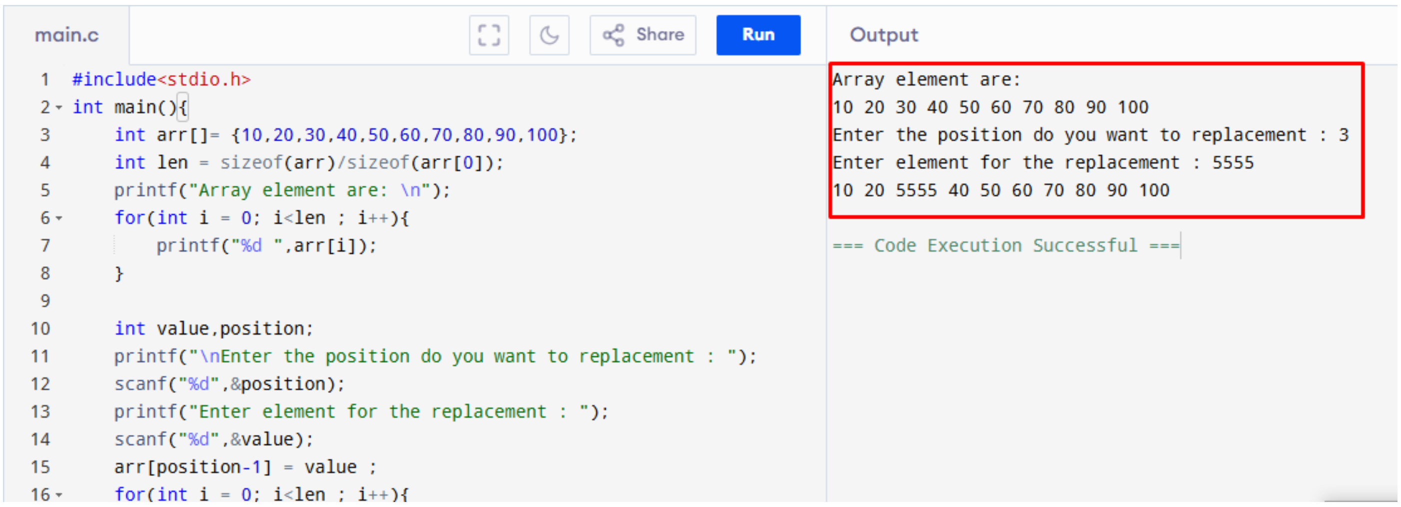
Course Teacher Name: Md. Abdul Hye Zebon

Designation: Lecturer

Submission Date: 12/12/24

Date: **14-09-2024**
Problem No: **01**
Problem Statement: **How to replace a specific element in an array.**

```
#include<stdio.h>
int main(){
    int arr[]= {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
    int len = sizeof(arr)/sizeof(arr[0]);
    printf("Array element are: \n");
    for(int i = 0; i<len ; i++){
        printf("%d ", arr[i]);
    }
    int value, position;
    printf("\nEnter the position do you want to replacement : ");
    scanf("%d", &position);
    printf("Enter element for the replacement : ");
    scanf("%d", &value);
    arr[position-1] = value ;
    for(int i = 0; i<len ; i++){
        printf("%d ", arr[i]);
    }
}
```

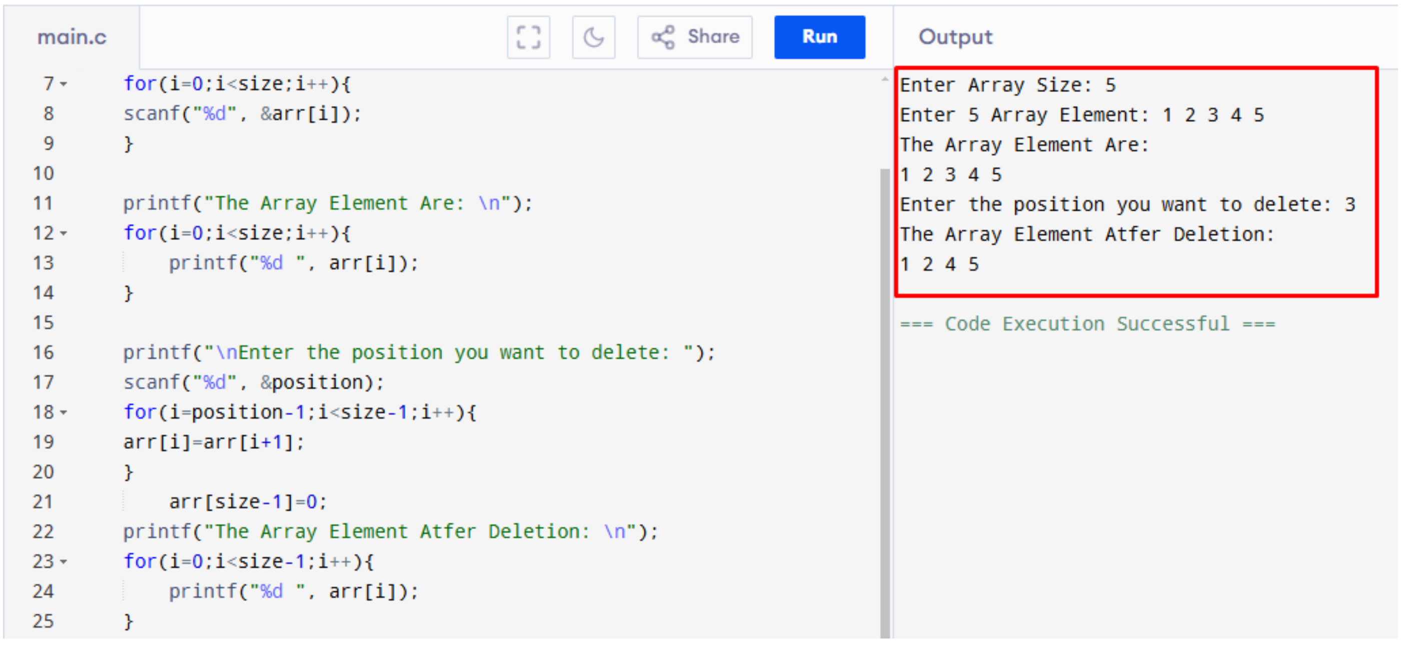


```
#include<stdio.h>
int main(){
    int arr[]= {10,20,30,40,50,60,70,80,90,100};
    int len = sizeof(arr)/sizeof(arr[0]);
    printf("Array element are: \n");
    for(int i = 0; i<len ; i++){
        printf("%d ",arr[i]);
    }

    int value,position;
    printf("\nEnter the position do you want to replacement : ");
    scanf("%d",&position);
    printf("Enter element for the replacement : ");
    scanf("%d",&value);
    arr[position-1] = value ;
    for(int i = 0; i<len ; i++){
```

```
Array element are:
10 20 30 40 50 60 70 80 90 100
Enter the position do you want to replacement : 3
Enter element for the replacement : 5555
10 20 5555 40 50 60 70 80 90 100

=== Code Execution Successful ===
```

Date: **14-09-2024**
Problem No: **02**
Problem Statement: **Delete Specific Element in the Array.**

```
#include <stdio.h>
int main() {
   int arr[100], i, size, position;
   printf("Enter Array Size: ");
   scanf("%d", &size);
   printf("Enter %d Array Element: ", size);
   for(i=0; i<size; i++){
   scanf("%d", &arr[i]);
   }
   printf("The Array Element Are: \n");
   for(i=0; i<size; i++){
      printf("%d ", arr[i]);
   }
   printf("\nEnter the position you want to delete: ");
   scanf("%d", &position);
   for(i=position-1;i<size-1;i++){
   arr[i]=arr[i+1];
   }
      arr[size-1]=0;
   printf("The Array Element Atfer Deletion: \n");
   for(i=0;i<size-1;i++){
      printf("%d ", arr[i]);
   }
   return 0;
}
```

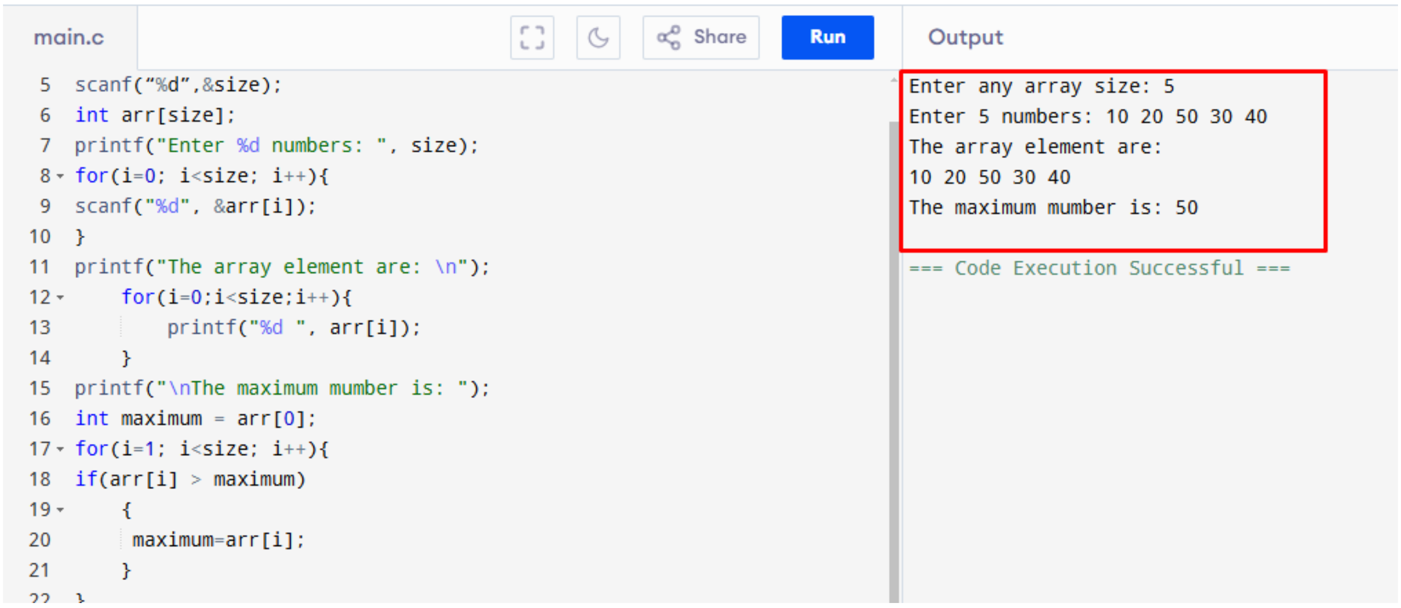


```c
    for(i=0;i<size;i++){
    scanf("%d", &arr[i]);
    }

    printf("The Array Element Are: \n");
    for(i=0;i<size;i++){
        printf("%d ", arr[i]);
    }

    printf("\nEnter the position you want to delete: ");
    scanf("%d", &position);
    for(i=position-1;i<size-1;i++){
    arr[i]=arr[i+1];
    }
        arr[size-1]=0;
    printf("The Array Element Atfer Deletion: \n");
    for(i=0;i<size-1;i++){
        printf("%d ", arr[i]);
    }
```

```
Enter Array Size: 5
Enter 5 Array Element: 1 2 3 4 5
The Array Element Are:
1 2 3 4 5
Enter the position you want to delete: 3
The Array Element Atfer Deletion:
1 2 4 5

=== Code Execution Successful ===
```

Date: **14-09-2024**
Problem No: **03**
Problem Statement: **Find the Maximum Number in the Array.**

```c
#include <stdio.h>

int main(){
int i, size;
printf("Enter any array size: ");
scanf("%d", &size);
int arr[size];
printf("Enter %d numbers: ", size);
for(i=0; i<size; i++){
scanf("%d", &arr[i]);
}
printf("The array element are: \n");
   for(i=0;i<size;i++){
      printf("%d ", arr[i]);
   }
printf("\nThe maximum mumber is: ");
int maximum = arr[0];
for(i=1; i<size; i++){
if(arr[i] > maximum)
   {
    maximum=arr[i];
   }
}
printf("%d", maximum);
return 0;
}
```

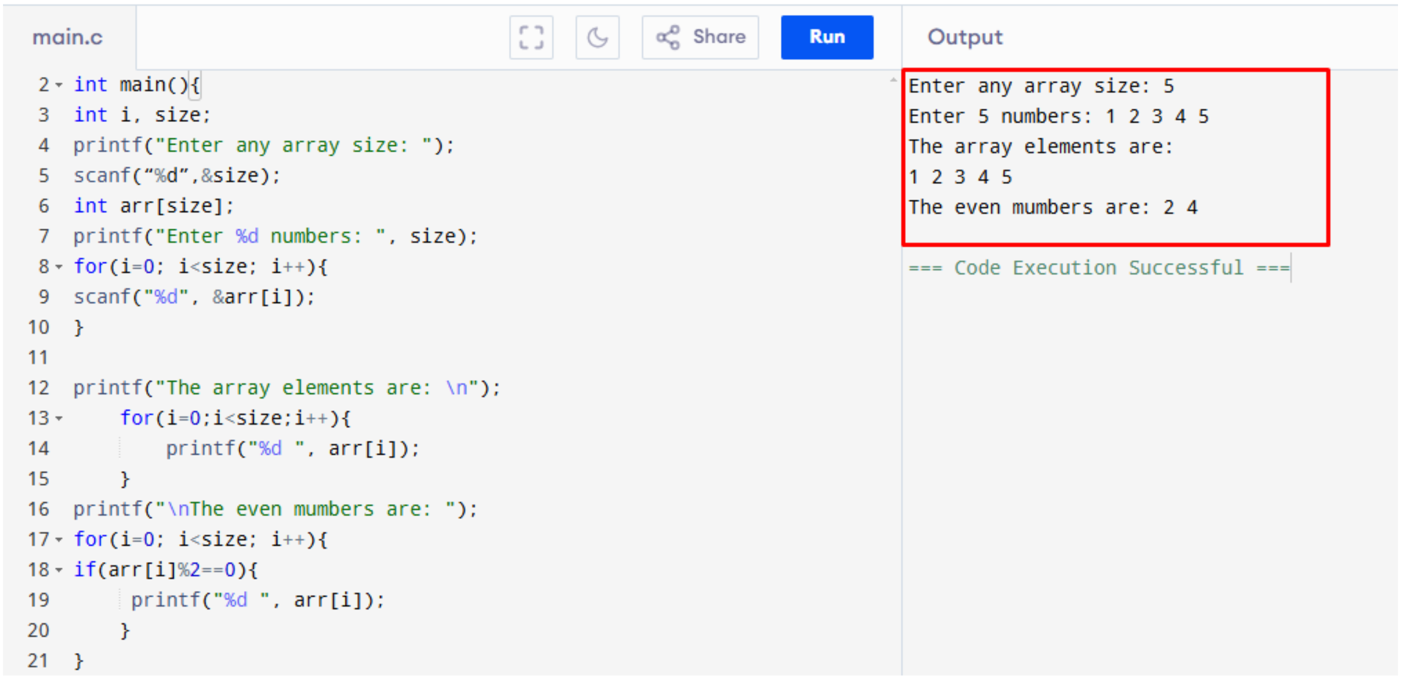


```
scanf("%d",&size);
int arr[size];
printf("Enter %d numbers: ", size);
for(i=0; i<size; i++){
scanf("%d", &arr[i]);
}
printf("The array element are: \n");
    for(i=0;i<size;i++){
        printf("%d ", arr[i]);
    }
printf("\nThe maximum mumber is: ");
int maximum = arr[0];
for(i=1; i<size; i++){
if(arr[i] > maximum)
    {
     maximum=arr[i];
    }
}
```

```
Enter any array size: 5
Enter 5 numbers: 10 20 50 30 40
The array element are:
10 20 50 30 40
The maximum mumber is: 50

=== Code Execution Successful ===
```

Date: **14-09-2024**
Problem No: **04**
Problem Statement: **Find the Even Numbers in the Array.**

```
#include <stdio.h>
int main(){
int i, size;
printf("Enter any array size: ");
scanf("%d",&size);
int arr[size];
printf("Enter %d numbers: ", size);
for(i=0; i<size; i++){
scanf("%d", &arr[i]);
}
printf("The array elements are: \n");
   for(i=0; i<size; i++){
      printf("%d ", arr[i]);
   }
printf("\nThe even mumbers are: ");
for(i=0; i<size; i++){
if(arr[i]%2==0){
    printf("%d ", arr[i]);
   }
}
return 0;
}
```

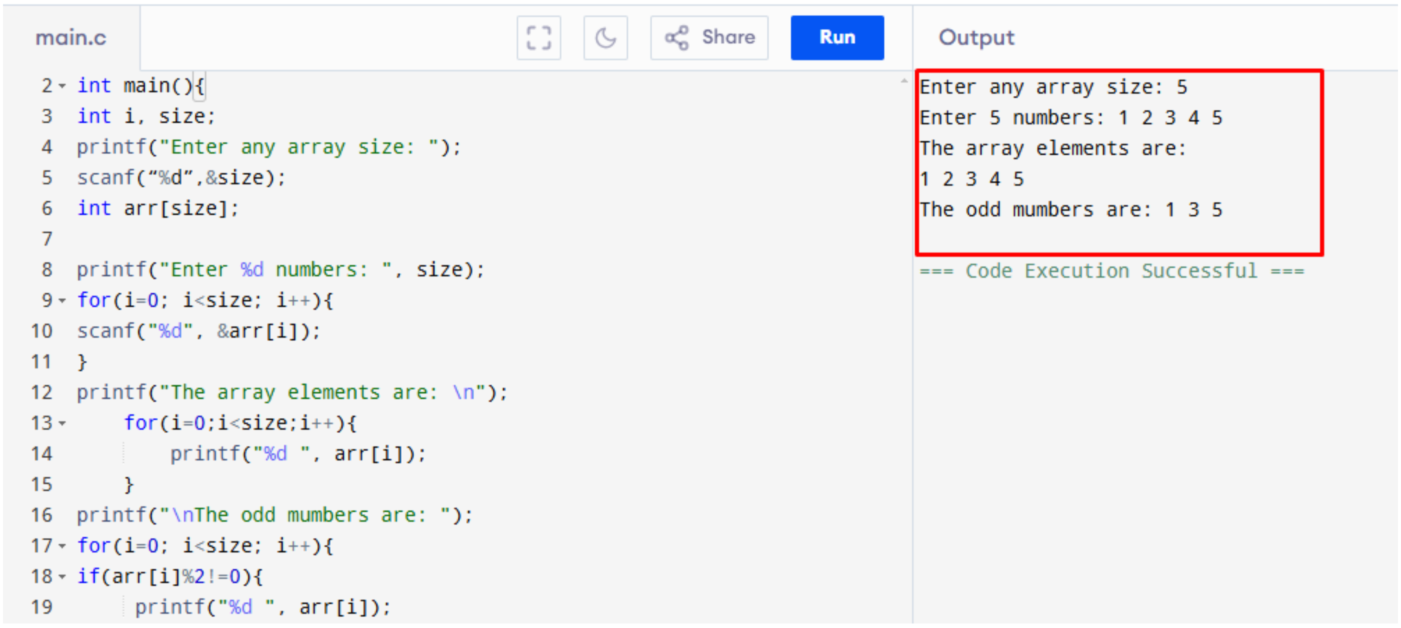


```
int main(){
int i, size;
printf("Enter any array size: ");
scanf("%d",&size);
int arr[size];
printf("Enter %d numbers: ", size);
for(i=0; i<size; i++){
scanf("%d", &arr[i]);
}

printf("The array elements are: \n");
    for(i=0;i<size;i++){
        printf("%d ", arr[i]);
    }
printf("\nThe even mumbers are: ");
for(i=0; i<size; i++){
if(arr[i]%2==0){
     printf("%d ", arr[i]);
    }
}
```

```
Enter any array size: 5
Enter 5 numbers: 1 2 3 4 5
The array elements are:
1 2 3 4 5
The even mumbers are: 2 4

=== Code Execution Successful ===
```

Date: **14-09-2024**
Problem No: **05**
Problem Statement: **Find the Odd Numbers in the Array.**

```
#include <stdio.h>
int main(){
int i, size;
printf("Enter any array size: ");
scanf(“%d”,&size);
int arr[size];

printf("Enter %d numbers: ", size);
for(i=0; i<size; i++){
scanf("%d", &arr[i]);
}
printf("The array elements are: \n");
   for(i=0; i<size; i++){
      printf("%d ", arr[i]);
   }
printf("\nThe odd mumbers are: ");
for(i=0; i<size; i++){
if(arr[i]%2!=0){
    printf("%d ", arr[i]);
   }
}
return 0;
}
```

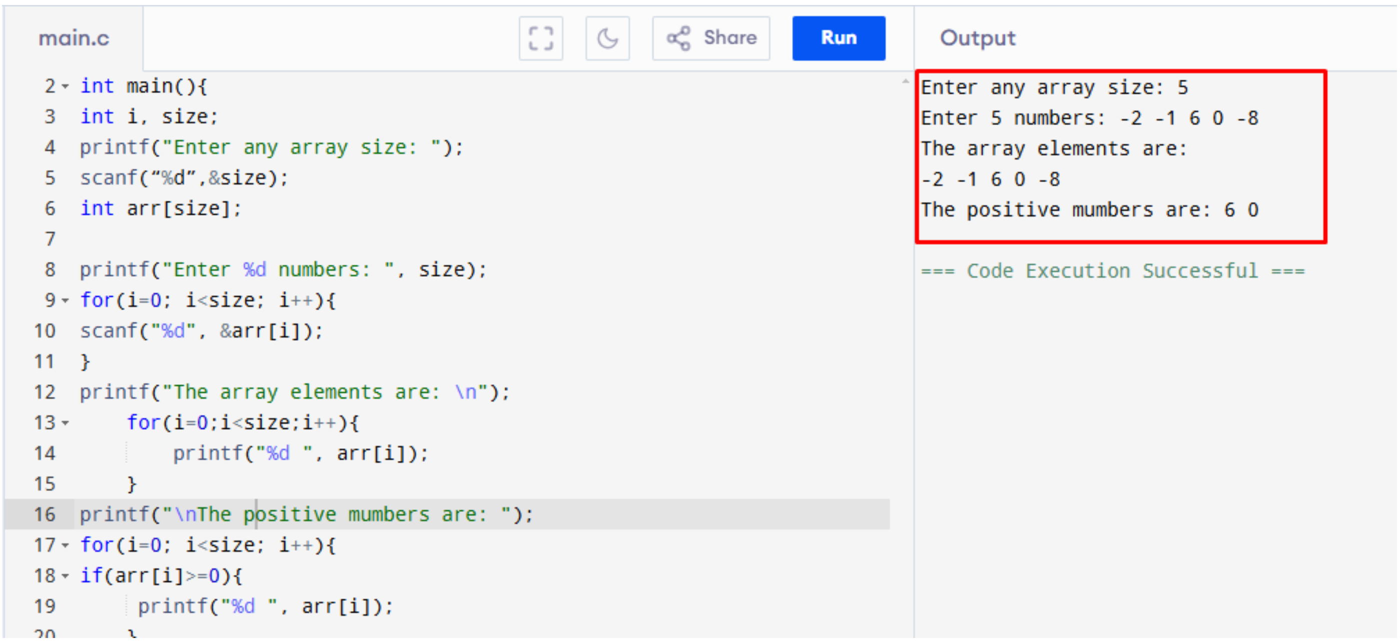

main.c Share Run Output

```
int main(){
int i, size;
printf("Enter any array size: ");
scanf("%d",&size);
int arr[size];

printf("Enter %d numbers: ", size);
for(i=0; i<size; i++){
scanf("%d", &arr[i]);
}
printf("The array elements are: \n");
    for(i=0;i<size;i++){
        printf("%d ", arr[i]);
    }
printf("\nThe odd numbers are: ");
for(i=0; i<size; i++){
if(arr[i]%2!=0){
      printf("%d ", arr[i]);
```

```
Enter any array size: 5
Enter 5 numbers: 1 2 3 4 5
The array elements are:
1 2 3 4 5
The odd numbers are: 1 3 5

=== Code Execution Successful ===
```

Date: **14-09-2024**
Problem No: **06**
Problem Statement: **Find the Positive Numbers in the Array.**

```
#include <stdio.h>
int main(){
int i, size;
printf("Enter any array size: ");
scanf("%d",&size);
int arr[size];


printf("Enter %d numbers: ", size);
for(i=0; i<size; i++){
scanf("%d", &arr[i]);
}
printf("The array elements are: \n");
    for(i=0; i<size; i++){
        printf("%d ", arr[i]);
    }
printf("\nThe positive mumbers are: ");
for(i=0; i<size; i++){
if(arr[i] >= 0){
    printf("%d ", arr[i]);
   }
}
return 0;
}
```


main.c
Share
Run
Output

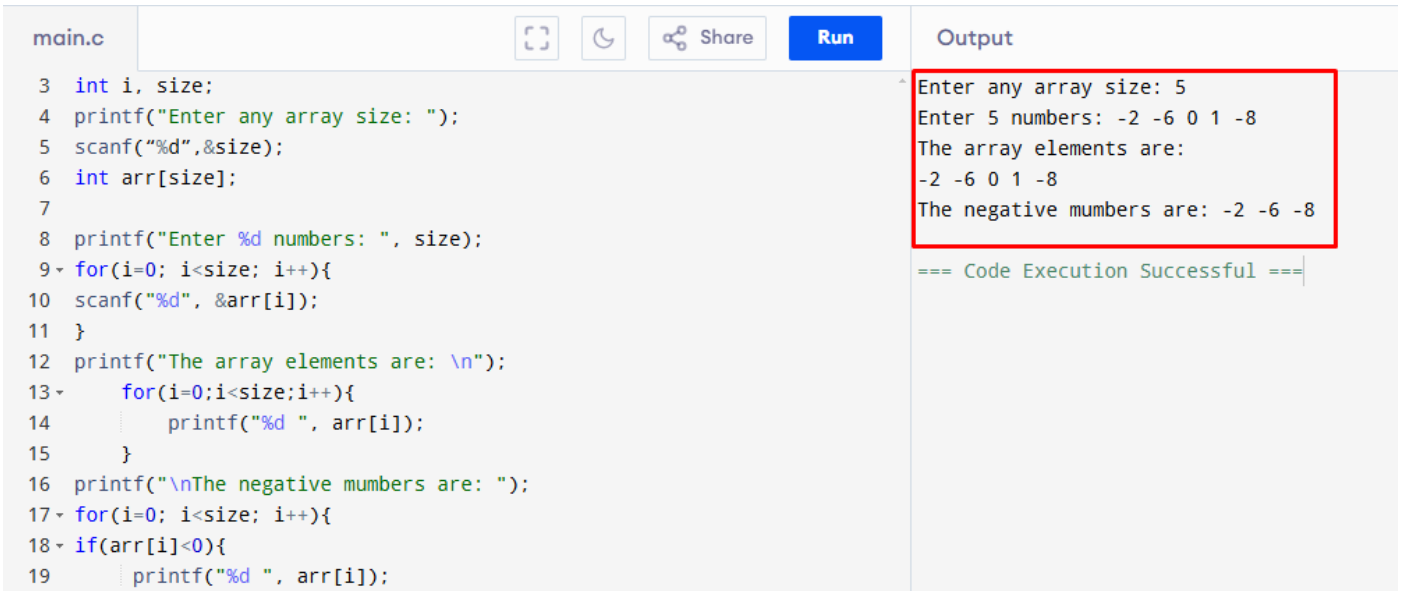
```
int main(){
int i, size;
printf("Enter any array size: ");
scanf("%d",&size);
int arr[size];

printf("Enter %d numbers: ", size);
for(i=0; i<size; i++){
scanf("%d", &arr[i]);
}
printf("The array elements are: \n");
    for(i=0;i<size;i++){
        printf("%d ", arr[i]);
    }
printf("\nThe positive mumbers are: ");
for(i=0; i<size; i++){
if(arr[i]>=0){
     printf("%d ", arr[i]);
```

```
Enter any array size: 5
Enter 5 numbers: -2 -1 6 0 -8
The array elements are:
-2 -1 6 0 -8
The positive mumbers are: 6 0

=== Code Execution Successful ===
```

Date: **14-09-2024**
Problem No: **07**
Problem Statement: **Find the Negative Numbers in the Array.**

```
#include <stdio.h>
int main(){
int i, size;
printf("Enter any array size: ");
scanf("%d",&size);
int arr[size];

printf("Enter %d numbers: ", size);
for(i=0; i<size; i++){
```

```
scanf("%d", &arr[i]);
}
printf("The array elements are: \n");
   for(i=0; i<size; i++){
      printf("%d ", arr[i]);
   }
printf("\nThe negative mumbers are: ");
for(i=0; i<size; i++){
if(arr[i]<0){
    printf("%d ", arr[i]);
   }
}
return 0;
}
```

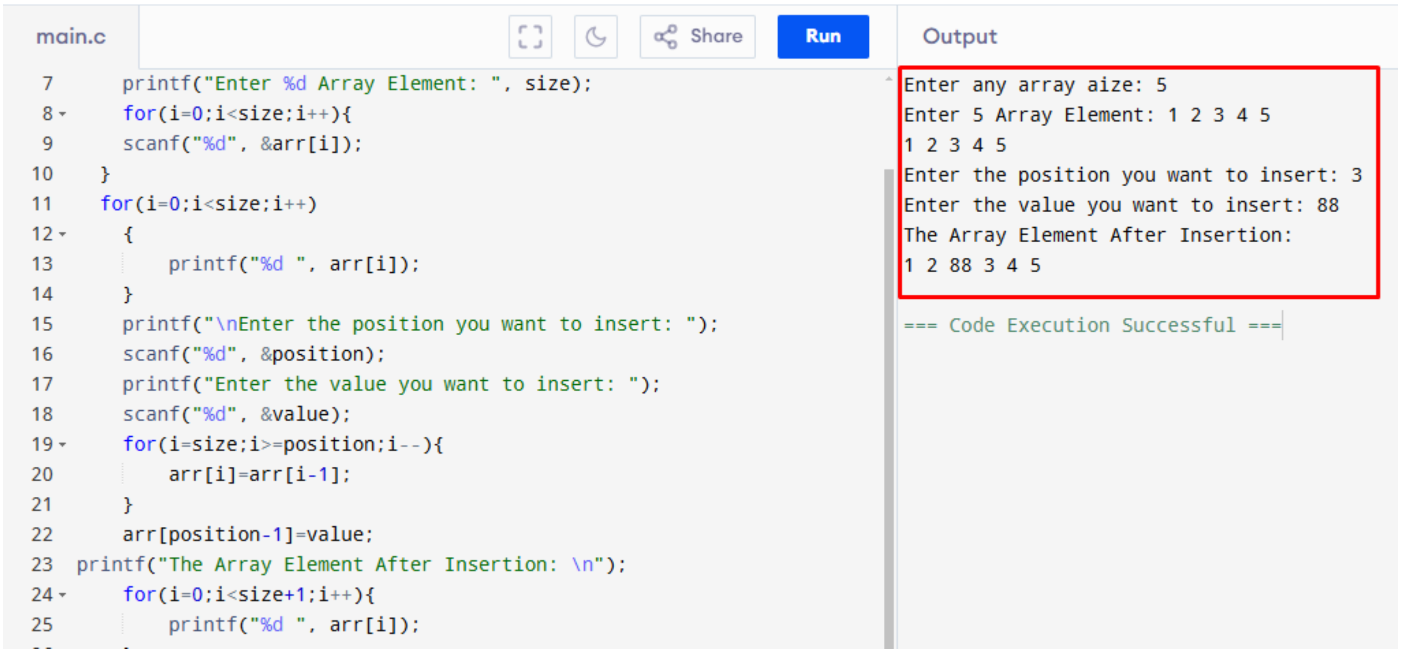


```
int i, size;
printf("Enter any array size: ");
scanf("%d",&size);
int arr[size];

printf("Enter %d numbers: ", size);
for(i=0; i<size; i++){
scanf("%d", &arr[i]);
}
printf("The array elements are: \n");
    for(i=0;i<size;i++){
        printf("%d ", arr[i]);
    }
printf("\nThe negative mumbers are: ");
for(i=0; i<size; i++){
if(arr[i]<0){
     printf("%d ", arr[i]);
```

```
Enter any array size: 5
Enter 5 numbers: -2 -6 0 1 -8
The array elements are:
-2 -6 0 1 -8
The negative mumbers are: -2 -6 -8

=== Code Execution Successful ===
```

Date: **21-09-2024**
Problem No: **08**
Problem Statement: **Insert a Number in the Array.**

```
#include <stdio.h>
int main()
{
  int arr[100], i, size, position, value;
  printf("Enter any array aize: ");
  scanf("%d", &size);
  printf("Enter %d Array Element: ", size);
  for(i=0;i<size;i++){
  scanf("%d", &arr[i]);
 }
 for(i=0; i<size; i++)
  {
    printf("%d ", arr[i]);
  }
  printf("\nEnter the position you want to insert: ");
  scanf("%d", &position);
  printf("Enter the value you want to insert: ");
  scanf("%d", &value);
  for(i=size; i>=position; i--){
    arr[i]=arr[i-1];
  }
  arr[position-1]=value;
printf("The Array Element After Insertion: \n");
  for(i=0; i<size+1; i++){
    printf("%d ", arr[i]);
  }
  return 0;
}
```



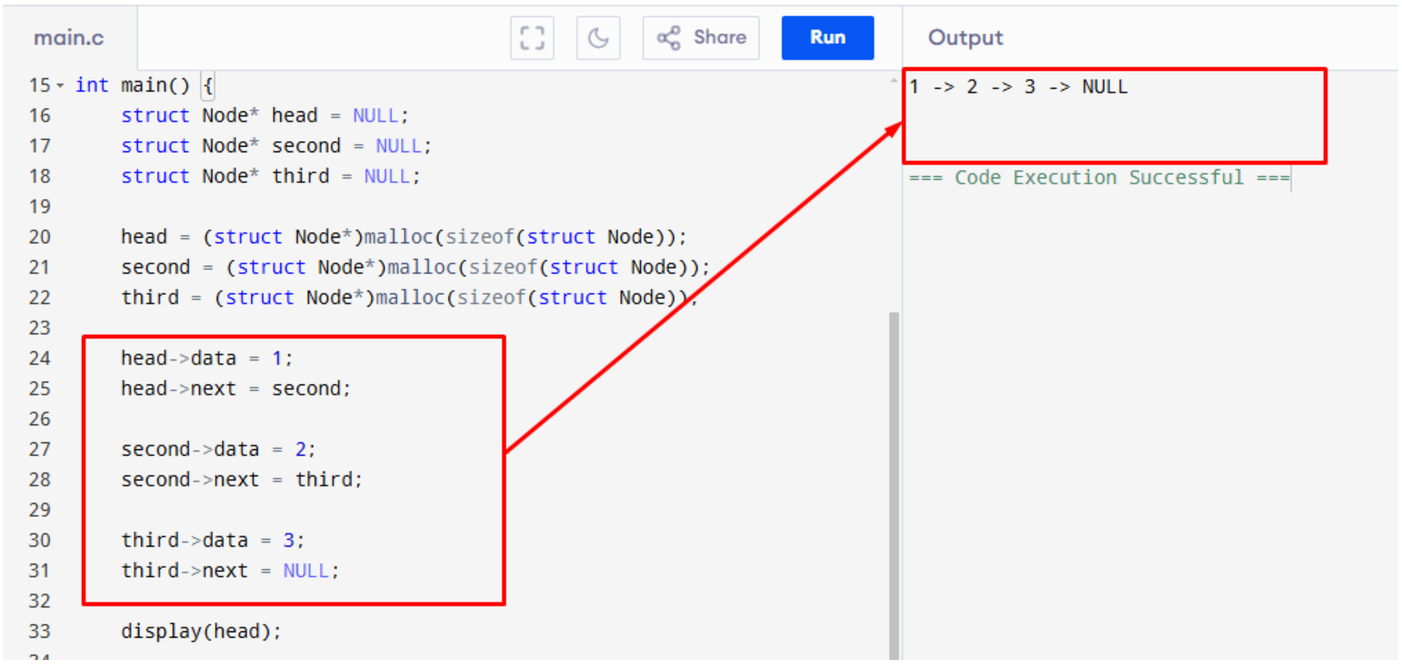
```c
    printf("Enter %d Array Element: ", size);
    for(i=0;i<size;i++){
    scanf("%d", &arr[i]);
   }
   for(i=0;i<size;i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\nEnter the position you want to insert: ");
    scanf("%d", &position);
    printf("Enter the value you want to insert: ");
    scanf("%d", &value);
    for(i=size;i>=position;i--){
        arr[i]=arr[i-1];
    }
    arr[position-1]=value;
  printf("The Array Element After Insertion: \n");
    for(i=0;i<size+1;i++){
        printf("%d ", arr[i]);
```

```
Enter any array aize: 5
Enter 5 Array Element: 1 2 3 4 5
1 2 3 4 5
Enter the position you want to insert: 3
Enter the value you want to insert: 88
The Array Element After Insertion:
1 2 88 3 4 5

=== Code Execution Successful ===
```

Date: **19-10-2024**
Problem No: **09**
Problem Statement: **LinkedList Creation Basic Concept and Manual Code.**

```c
#include <stdio.h>

#include <stdlib.h>

struct Node {

   int data;

   struct Node* next;

};

void display(struct Node* head) {

   struct Node* temp = head;

   while (temp != NULL) {

      printf("%d -> ", temp->data);

      temp = temp->next;

   }

   printf("NULL\n");

}

int main() {

   struct Node* head = NULL;
```

```
    struct Node* second = NULL;
    struct Node* third = NULL;
    head = (struct Node*)malloc(sizeof(struct Node));
    second = (struct Node*)malloc(sizeof(struct Node));
    third = (struct Node*)malloc(sizeof(struct Node));
    head->data = 1;
    head->next = second;
    second->data = 2;
    second->next = third;
    third->data = 3;
    third->next = NULL;
    display(head);
    return 0;
}
```

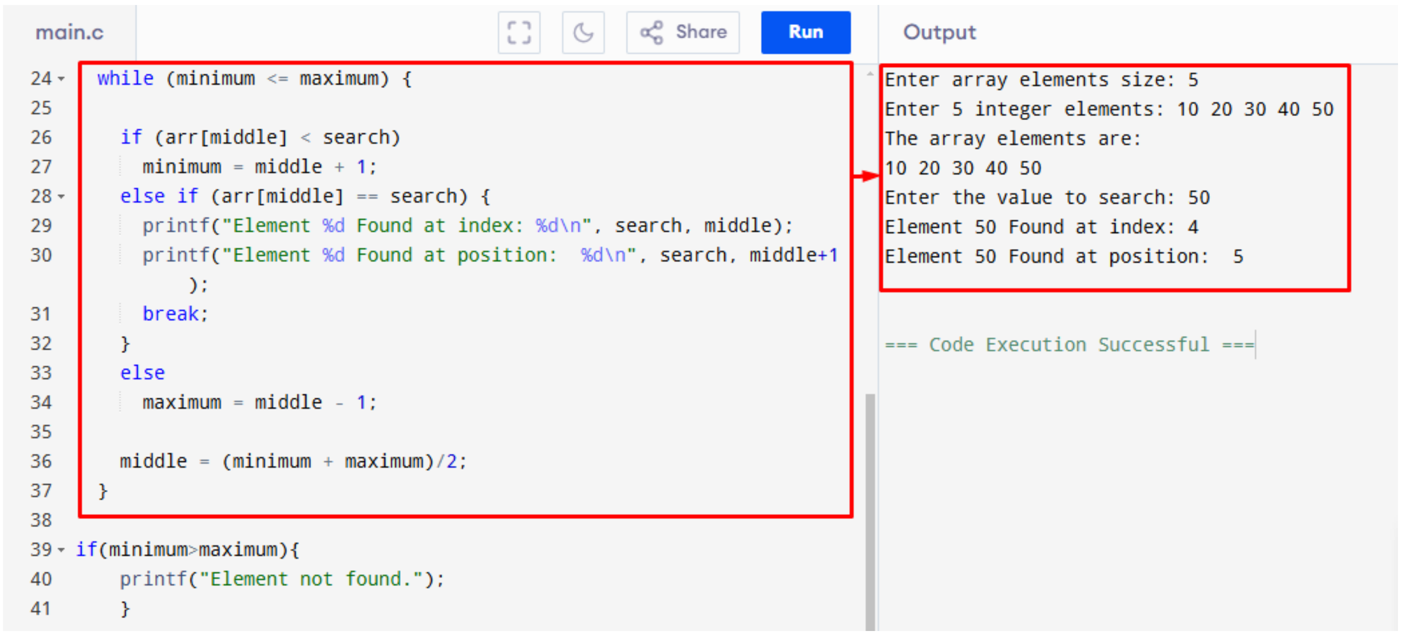


```
int main() {
    struct Node* head = NULL;
    struct Node* second = NULL;
    struct Node* third = NULL;

    head = (struct Node*)malloc(sizeof(struct Node));
    second = (struct Node*)malloc(sizeof(struct Node));
    third = (struct Node*)malloc(sizeof(struct Node));

    head->data = 1;
    head->next = second;

    second->data = 2;
    second->next = third;

    third->data = 3;
    third->next = NULL;

    display(head);
```

```
1 -> 2 -> 3 -> NULL

=== Code Execution Successful ===
```

Date: **19-10-2024**
Problem No: **10**
Problem Statement: Binary searching algorithm in an Array**.**

```
#include <stdio.h>
int main() {
  int i, minimum, maximum, middle, size, search, arr[100];


  printf("Enter array elements size: ");
  scanf("%d", &size);


  printf("Enter %d integer elements: ", size);


  for (i=0; i<size; i++){
    scanf("%d",&arr[i]);
}
printf("The array elements are: \n");
    for(i=0; i<size; i++){
        printf("%d ", arr[i]);
    }
  printf("\nEnter the value to search: ");
  scanf("%d", &search);
  minimum = 0;
  maximum = size - 1;
  middle = (minimum+maximum)/2;
  while (minimum <= maximum) {
    if (arr[middle] < search)
      minimum = middle + 1;
    else if (arr[middle] == search) {
      printf("Element %d Found at index: %d\n", search, middle);
      printf("Element %d Found at position:  %d\n", search, middle+1);
      break;
    }
    else
      maximum = middle - 1;
    middle = (minimum + maximum)/2;
  }
if(minimum>maximum){
    printf("Element not found.");
    }
  return 0;
}
```


main.c

```
while (minimum <= maximum) {

  if (arr[middle] < search)
    minimum = middle + 1;
  else if (arr[middle] == search) {
    printf("Element %d Found at index: %d\n", search, middle);
    printf("Element %d Found at position:  %d\n", search, middle+1
        );
    break;
  }
  else
    maximum = middle - 1;

  middle = (minimum + maximum)/2;
}

if(minimum>maximum){
    printf("Element not found.");
    }
```

Output

```
Enter array elements size: 5
Enter 5 integer elements: 10 20 30 40 50
The array elements are:
10 20 30 40 50
Enter the value to search: 50
Element 50 Found at index: 4
Element 50 Found at position:  5

=== Code Execution Successful ===
```

Date: **19-10-2024**
Problem No: **11**
Problem Statement: **Linear Search in Linked List with Multiple Occurrences.**

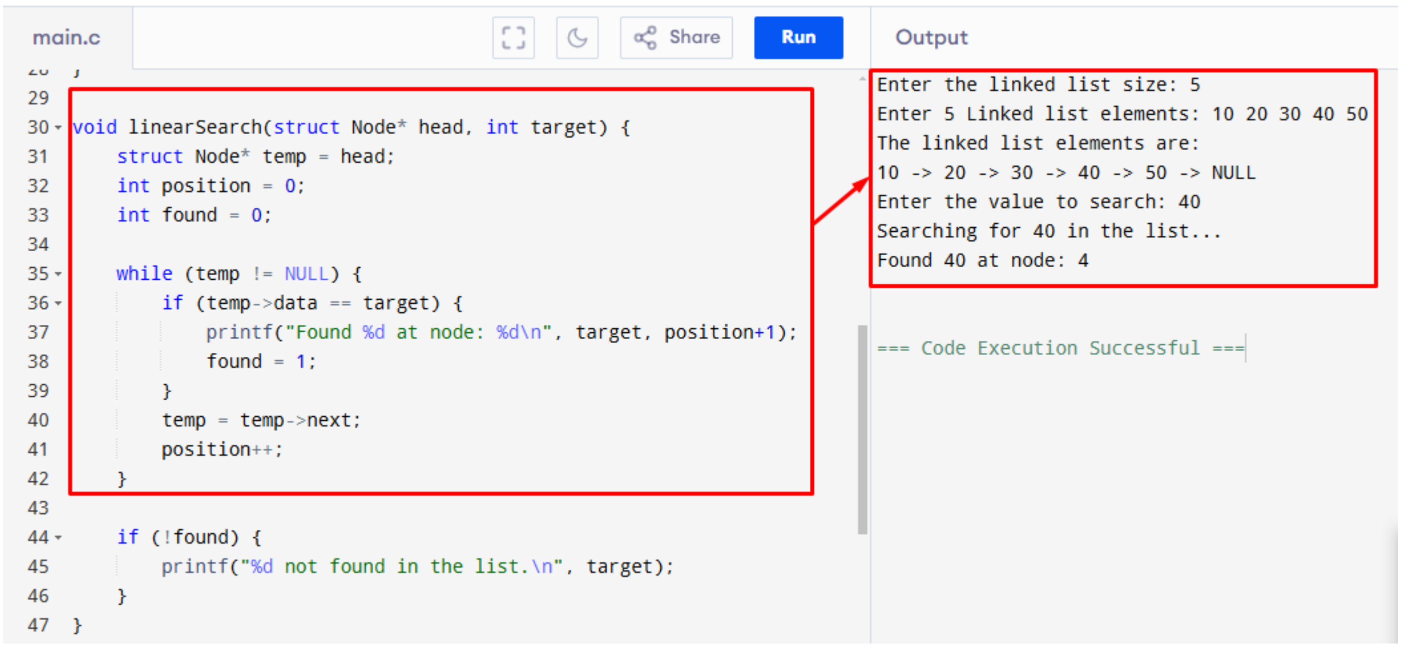
```
#include <stdio.h>

#include <stdlib.h>

struct Node {

int data;

struct Node* next; };

struct Node* createNode(int data) {

struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));

newNode->data = data;

newNode->next = NULL;

return newNode; }
```

```
void appendNode(struct Node** head, int data) {

struct Node* newNode = createNode(data);

if (*head == NULL) {

 *head = newNode;

return; }

struct Node* temp = *head;
while (temp->next != NULL) {
    temp = temp->next;
 }
temp->next = newNode;

}

void linearSearch(struct Node* head, int target) {

struct Node* temp = head;

int position = 0;

int found = 0;

while (temp != NULL) {
    if (temp->data == target) {
        printf("Found %d at node: %d\n", target, position+1);
        found = 1;
    }
    temp = temp->next;
    position++;
}

if (!found) {
    printf("%d not found in the list.\n", target);
}
}

void printList(struct Node* head) {

struct Node* temp = head; while (temp != NULL) {

printf("%d -> ", temp->data); temp = temp->next; }

printf("NULL\n"); }

int main() {

struct Node* head = NULL;
```

int size, value, target;

```
printf("Enter the linked list size: ");
scanf("%d", &size);

printf("Enter %d Linked list elements: ", size);
for (int i = 0; i < size; i++) {
    scanf("%d", &value);
    appendNode(&head, value);
}

printf("The linked list elements are: \n");
printList(head);

printf("Enter the value to search: ");
scanf("%d", &target);

printf("Searching for %d in the list...\n", target);
linearSearch(head, target);

return 0;
```

}

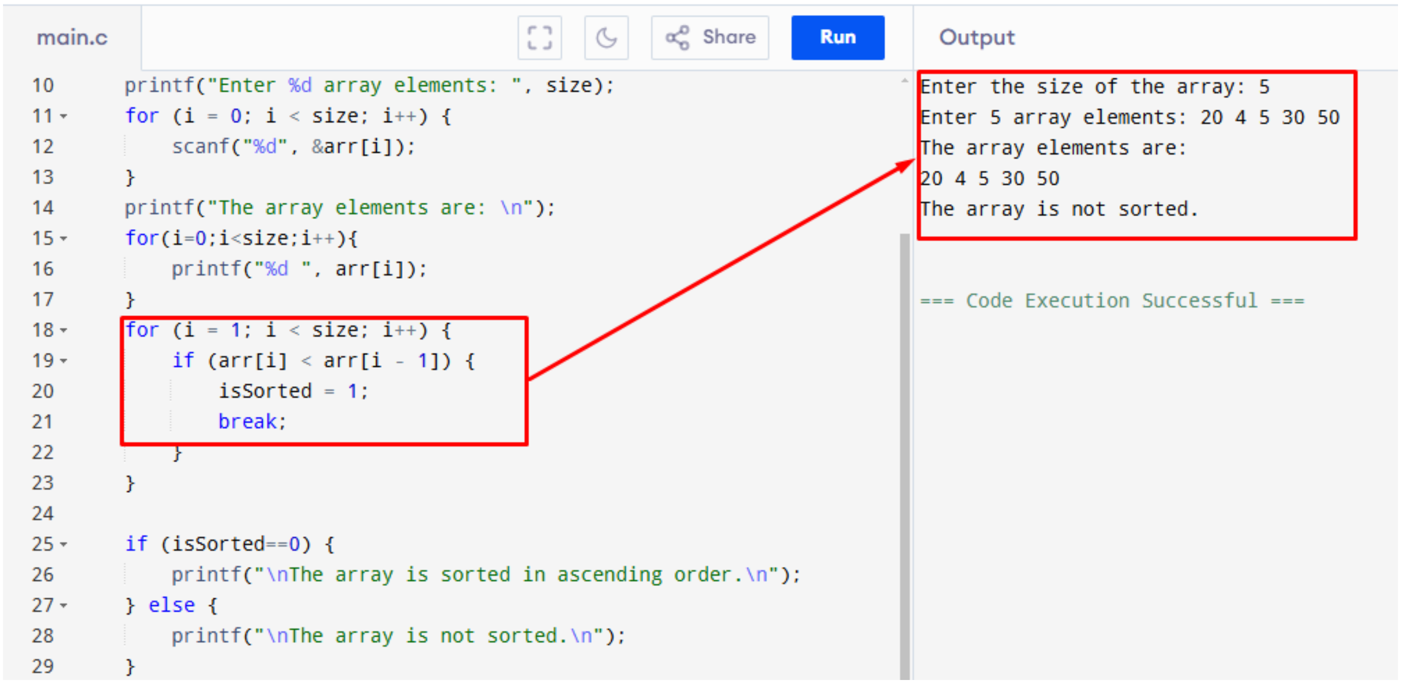


```
}

void linearSearch(struct Node* head, int target) {
    struct Node* temp = head;
    int position = 0;
    int found = 0;

    while (temp != NULL) {
        if (temp->data == target) {
            printf("Found %d at node: %d\n", target, position+1);
            found = 1;
        }
        temp = temp->next;
        position++;
    }

    if (!found) {
        printf("%d not found in the list.\n", target);
    }
}
```

```
Enter the linked list size: 5
Enter 5 Linked list elements: 10 20 30 40 50
The linked list elements are:
10 -> 20 -> 30 -> 40 -> 50 -> NULL
Enter the value to search: 40
Searching for 40 in the list...
Found 40 at node: 4

=== Code Execution Successful ===
```

Date: **26-10-2024**
Problem No: **12**
Problem Statement: **Check an array is sorted or not.**

```
#include <stdio.h>
int main() {
    int i, size, isSorted = 0;
    printf("Enter the size of the array: ");
    scanf("%d", &size);
    int arr[size];
    printf("Enter %d array elements: ", size);
    for (i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
    }
    printf("The array elements are: \n");
    for(i=0;i<size;i++){
        printf("%d ", arr[i]);
    }
    for (i = 1; i < size; i++) {
        if (arr[i] < arr[i - 1]) {
            isSorted = 1;
            break;
        }
    }
    if (isSorted==0) {
        printf("\nThe array is sorted in ascending order.\n");
    } else {
        printf("\nThe array is not sorted.\n");
    }
    return 0;
}
```



```c
    printf("Enter %d array elements: ", size);
    for (i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
    }
    printf("The array elements are: \n");
    for(i=0;i<size;i++){
        printf("%d ", arr[i]);
    }
    for (i = 1; i < size; i++) {
        if (arr[i] < arr[i - 1]) {
            isSorted = 1;
            break;
        }
    }

    if (isSorted==0) {
        printf("\nThe array is sorted in ascending order.\n");
    } else {
        printf("\nThe array is not sorted.\n");
    }
```

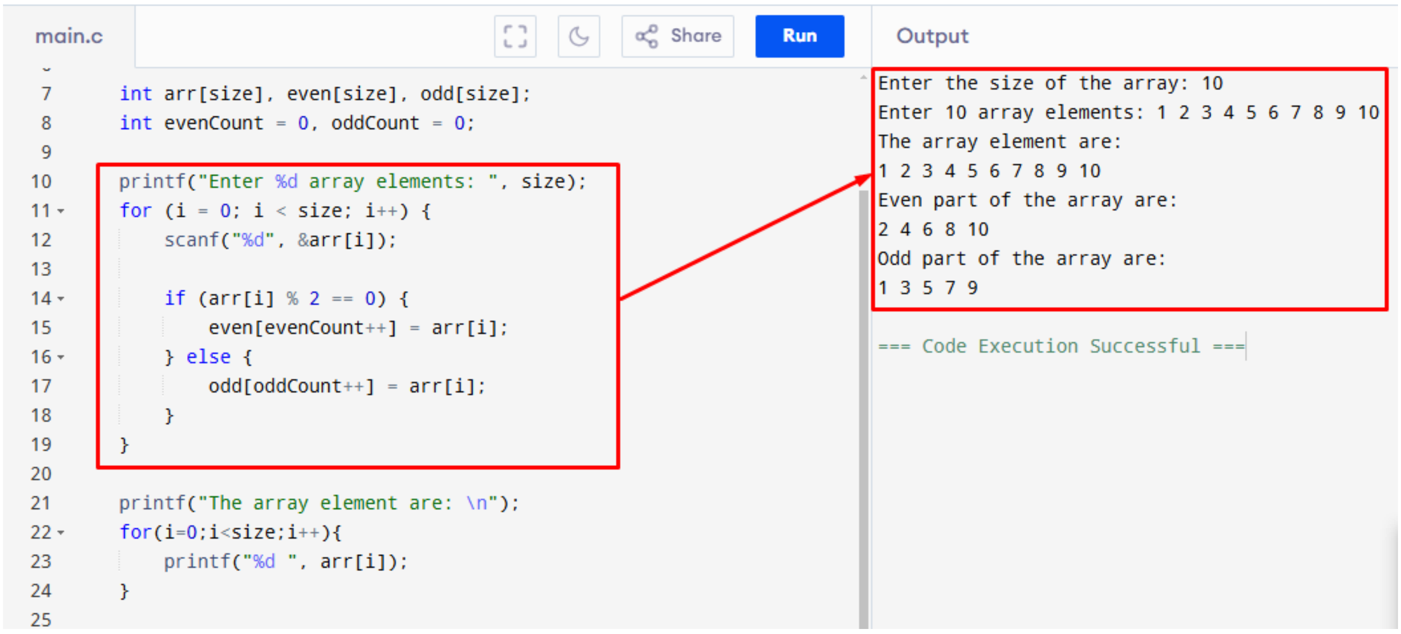
```
Enter the size of the array: 5
Enter 5 array elements: 20 4 5 30 50
The array elements are:
20 4 5 30 50
The array is not sorted.


=== Code Execution Successful ===
```

Date: **26-10-2024**

Problem No: **13**

Problem Statement: **Split an Array into Two Parts Even First Part and Odd Elements Second Part.**

```c
#include <stdio.h>
int main() {
    int i, size;
    printf("Enter the size of the array: ");
    scanf("%d", &size);

    int arr[size], even[size], odd[size];
    int evenCount = 0, oddCount = 0;

    printf("Enter %d array elements: ", size);
    for (i = 0; i < size; i++) {
        scanf("%d", &arr[i]);

        if (arr[i] % 2 == 0) {
            even[evenCount++] = arr[i];
        } else {
            odd[oddCount++] = arr[i];
        }
    }

    printf("The array element are: \n");
    for(i=0;i<size;i++){
        printf("%d ", arr[i]);
    }
    printf("\nEven part of the array are:\n");
    for (i = 0; i < evenCount; i++) {
        printf("%d ", even[i]);
    }
    printf("\nOdd part of the array are:\n");
    for (i = 0; i < oddCount; i++) {
        printf("%d ", odd[i]);
    }
    return 0;
}
```



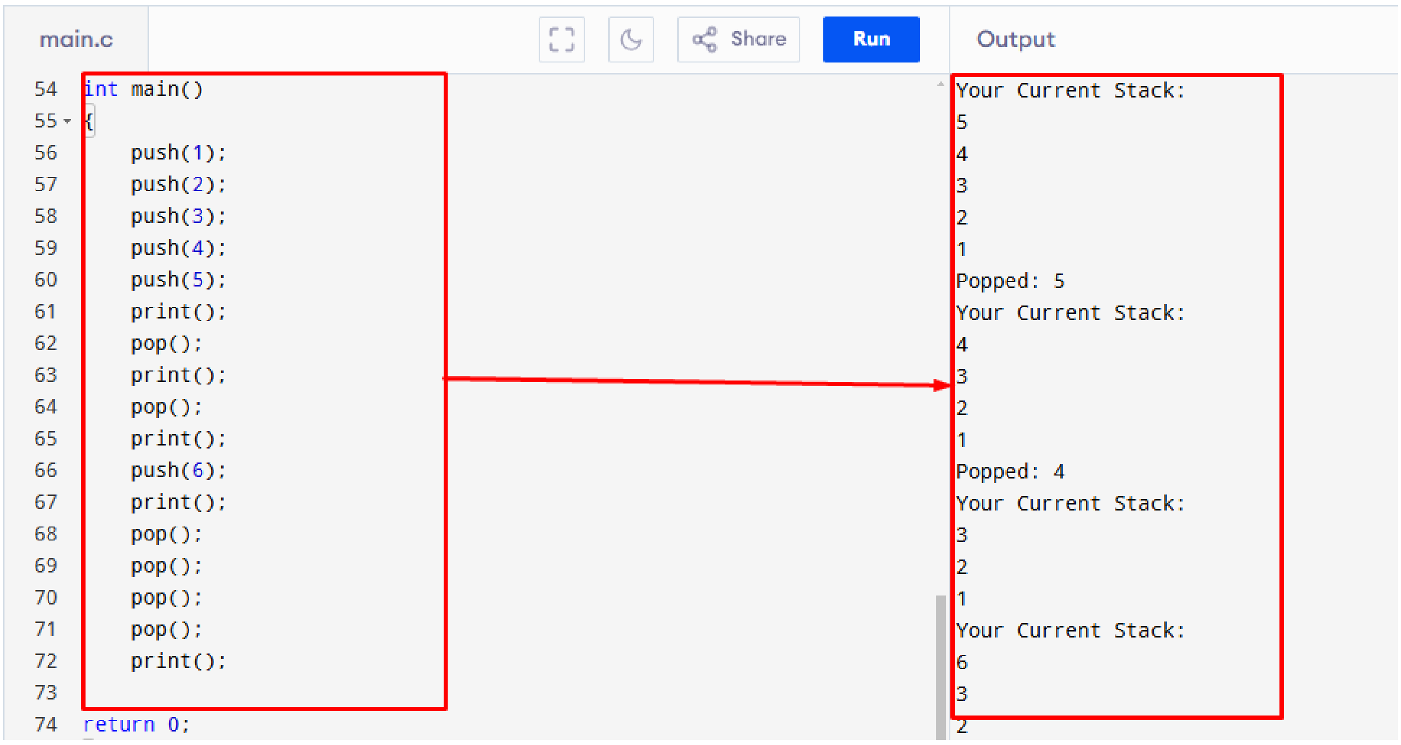
```
    int arr[size], even[size], odd[size];
    int evenCount = 0, oddCount = 0;

    printf("Enter %d array elements: ", size);
    for (i = 0; i < size; i++) {
        scanf("%d", &arr[i]);

        if (arr[i] % 2 == 0) {
            even[evenCount++] = arr[i];
        } else {
            odd[oddCount++] = arr[i];
        }
    }

    printf("The array element are: \n");
    for(i=0;i<size;i++){
        printf("%d ", arr[i]);
    }

```

```
Enter the size of the array: 10
Enter 10 array elements: 1 2 3 4 5 6 7 8 9 10
The array element are:
1 2 3 4 5 6 7 8 9 10
Even part of the array are:
2 4 6 8 10
Odd part of the array are:
1 3 5 7 9

=== Code Execution Successful ===
```

Date: **01-11-2024**
Problem No: **14**
Problem Statement: **Implement a Stack to Support Efficient Creation, Data insertion, and Deletion Operations.**

```
#include<stdio.h>
int top = -1;
int stack[5];
int isFull(){
   if (top==4){
      return 1;
   }else{
      return 0;
   } }
int isEmpty(){
   if(top == -1){
      return 1;
   }else{
      return 0;
   }
}
void push(int data){
   if (isFull()){
      printf("Stack Overflow!\n");
      return;
   }
   top = top+1;
   stack[top] = data;
}
void pop(){
   if(isEmpty()) {
      printf("Stack is empty!\n");
```

```
        return;
    }
    printf("Popped: %d\n",stack[top]);
    top=top-1;
}
void print(){
    if(isEmpty()){
        printf("Empty Stack");
        return;
    }
    printf("Your Current Stack:\n");
    for(int i=top;i>=0;i--){
        printf("%d\n", stack[i]);
    }
}
int main(){
    push(1);
    push(2);
    push(3);
    push(4);
    push(5);
    print();
    pop();
    print();
    pop();
    print();
    push(6);
    print();
    pop();
    pop();
    pop();
    pop();
    print();
return 0;
}
```

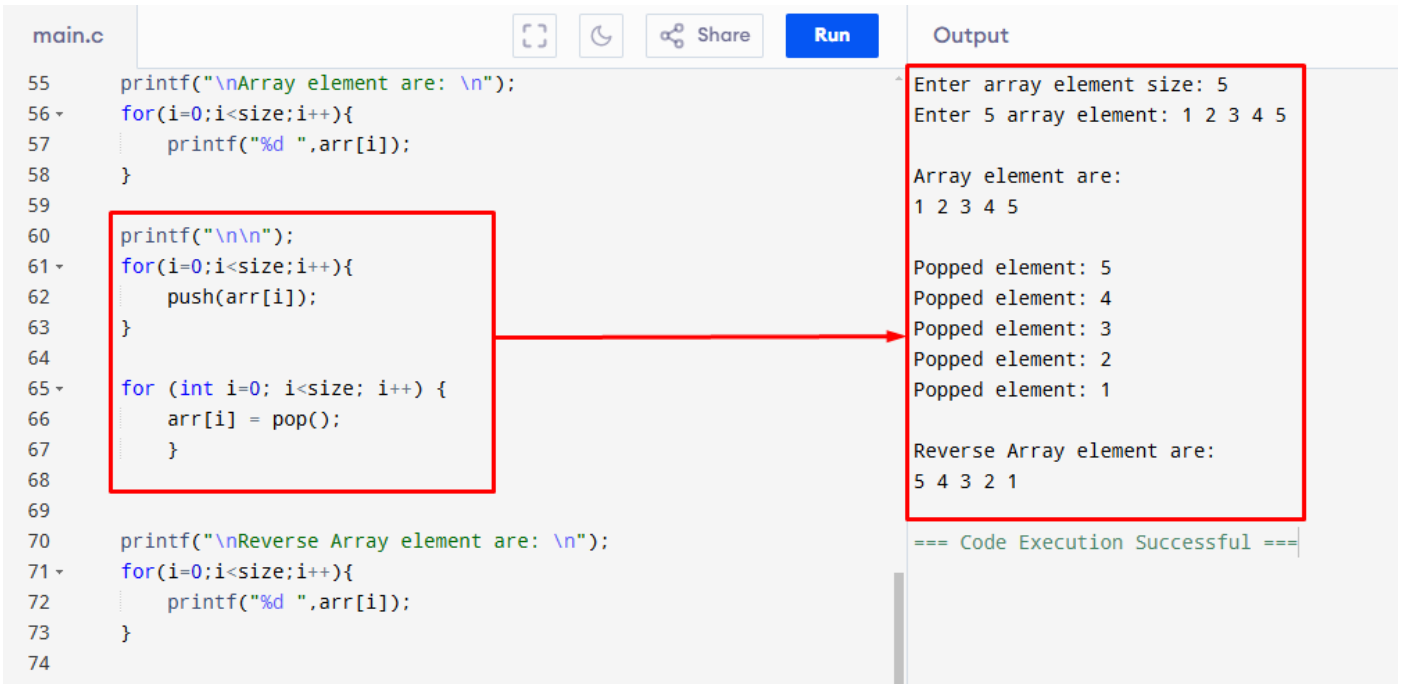


```
int main()
{
    push(1);
    push(2);
    push(3);
    push(4);
    push(5);
    print();
    pop();
    print();
    pop();
    print();
    push(6);
    print();
    pop();
    pop();
    pop();
    pop();
    print();

return 0;
```

```
Your Current Stack:
5
4
3
2
1
Popped: 5
Your Current Stack:
4
3
2
1
Popped: 4
Your Current Stack:
3
2
1
Your Current Stack:
6
3
2
```

Date: **02-11-2024**
Problem No: **15**
Problem Statement: **Reverse an Array Using a Stack.**

```
#include<stdio.h>
#define stack_size 100
int top = -1;
int stack[stack_size];
int isFull(){
    if (top==stack_size-1){
        return 1;
    }else{
        return 0;
    } }
int isEmpty(){
    if(top == -1){
        return 1;
    }
    else{
        return 0;
    }
}
void push(int data){
    if (isFull()){
        printf("Stack Overflow!\n");
        return;
    }
    top++;
    stack[top] = data;
}
int pop(){
    if(isEmpty()){
        printf("Stack is empty!\n");
        return 1;
    }
    printf("Popped element: %d\n",stack[top]);
    return stack[top--];
}
int main(){
    int i,size;
    printf("Enter array element size: ");
    scanf("%d",&size);
    int arr[size];
    printf("Enter %d array element: ", size);
    for(i=0;i<size;i++){
        scanf("%d",&arr[i]);
    }
    printf("\nArray element are: \n");
    for(i=0;i<size;i++){
        printf("%d ",arr[i]);
    }
    printf("\n\n");
    for(i=0;i<size;i++){
        push(arr[i]);
    }
    for (int i=0; i<size; i++) {
        arr[i] = pop();
        }
    printf("\nReverse Array element are: \n");
    for(i=0;i<size;i++){
        printf("%d ",arr[i]);
    }
return 0;
}
```

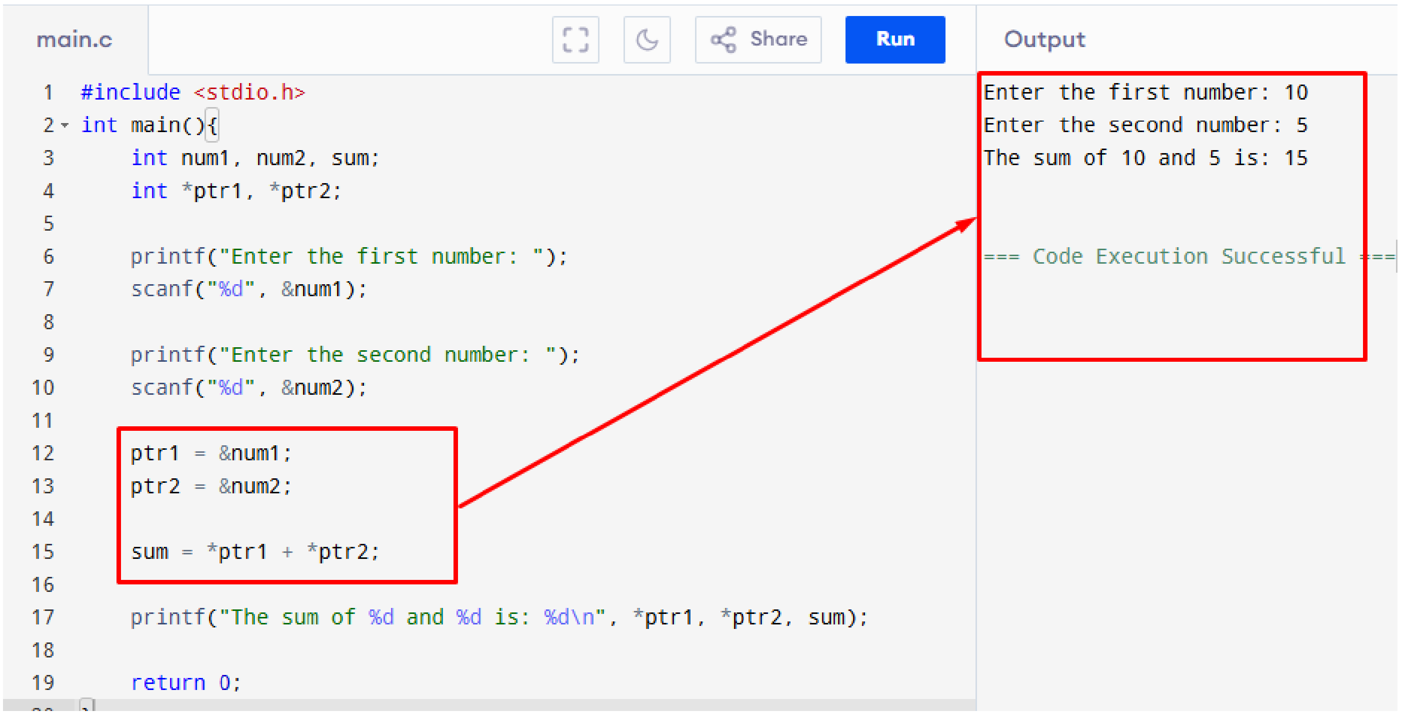


```
    printf("\nArray element are: \n");
    for(i=0;i<size;i++){
        printf("%d ",arr[i]);
    }

    printf("\n\n");
    for(i=0;i<size;i++){
        push(arr[i]);
    }

    for (int i=0; i<size; i++) {
        arr[i] = pop();
        }


    printf("\nReverse Array element are: \n");
    for(i=0;i<size;i++){
        printf("%d ",arr[i]);
    }

```

```
Enter array element size: 5
Enter 5 array element: 1 2 3 4 5

Array element are:
1 2 3 4 5

Popped element: 5
Popped element: 4
Popped element: 3
Popped element: 2
Popped element: 1

Reverse Array element are:
5 4 3 2 1

=== Code Execution Successful ===
```

Date: **09-11-2024**
Problem No: **16**
Problem Statement: **Sum of Two Numbers Using Pointers.**

```
#include <stdio.h>

int main(){

int num1, num2, sum; int *ptr1, *ptr2;

printf("Enter the first number: ");
scanf("%d", &num1);

printf("Enter the second number: ");
scanf("%d", &num2);

ptr1 = &num1;
ptr2 = &num2;

sum = *ptr1 + *ptr2;

printf("The sum of %d and %d is: %d\n", *ptr1, *ptr2, sum);

return 0;

}
```

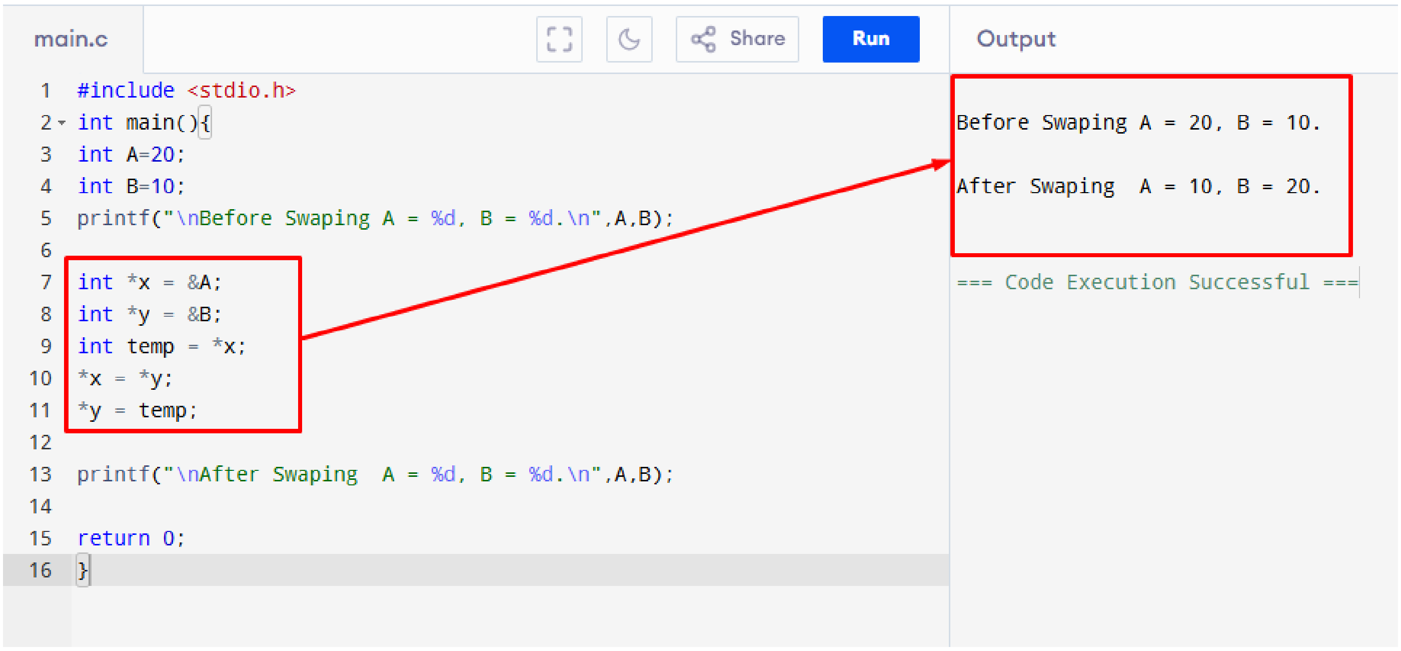


```c
#include <stdio.h>
int main(){
    int num1, num2, sum;
    int *ptr1, *ptr2;

    printf("Enter the first number: ");
    scanf("%d", &num1);

    printf("Enter the second number: ");
    scanf("%d", &num2);

    ptr1 = &num1;
    ptr2 = &num2;

    sum = *ptr1 + *ptr2;

    printf("The sum of %d and %d is: %d\n", *ptr1, *ptr2, sum);

    return 0;
```

```
Enter the first number: 10
Enter the second number: 5
The sum of 10 and 5 is: 15


=== Code Execution Successful ===
```

Date: **09-11-2024**
Problem No: **17**
Problem Statement: **Swap Two Numbers Using Pointers With Temp Variable.**

```c
#include <stdio.h>

int main(){

int A=20; int B=10;

 printf("\nBefore Swaping A = %d, B = %d.\n",A,B);

int *x = &A;

int *y = &B;

int temp = *x;

*x = *y;

*y = temp;

printf("\nAfter Swaping A = %d, B = %d.\n",A,B);

return 0;

}
```

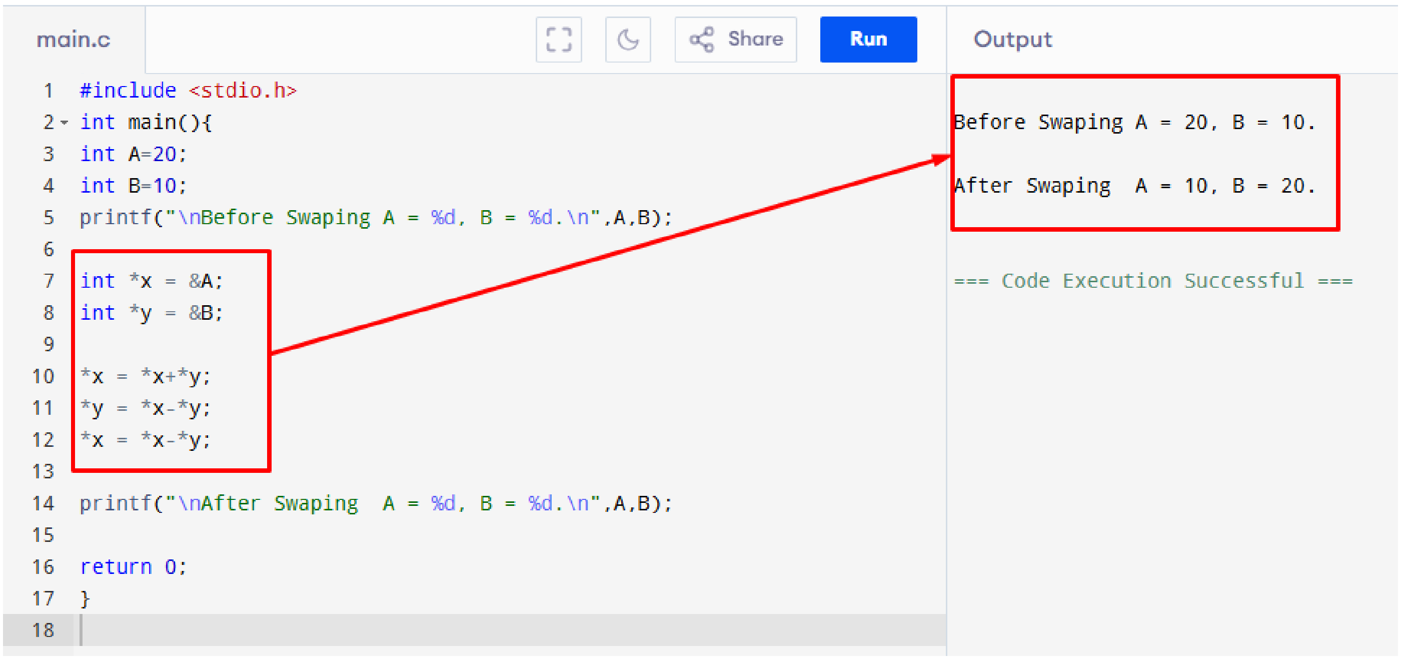


```
#include <stdio.h>
int main(){
int A=20;
int B=10;
printf("\nBefore Swaping A = %d, B = %d.\n",A,B);

int *x = &A;
int *y = &B;
int temp = *x;
*x = *y;
*y = temp;

printf("\nAfter Swaping  A = %d, B = %d.\n",A,B);

return 0;
}
```

```
Before Swaping A = 20, B = 10.

After Swaping  A = 10, B = 20.

=== Code Execution Successful ===
```

Date: **09-11-2024**
Problem No: **18**
Problem Statement: **Swap Two Numbers Using Pointers Without Temp Variable.**

```
#include <stdio.h>

int main(){

int A=20;

int B=10;

printf("\nBefore Swaping A = %d, B = %d.\n",A,B);

int *x = &A;

int *y = &B;

*x = *x+*y;

*y = *x-*y;

*x = *x-*y;

printf("\nAfter Swaping A = %d, B = %d.\n",A,B);

return 0;

}
```


main.c

Share

Run

Output

```
#include <stdio.h>
int main(){
int A=20;
int B=10;
printf("\nBefore Swaping A = %d, B = %d.\n",A,B);

int *x = &A;
int *y = &B;

*x = *x+*y;
*y = *x-*y;
*x = *x-*y;

printf("\nAfter Swaping  A = %d, B = %d.\n",A,B);

return 0;
}
```

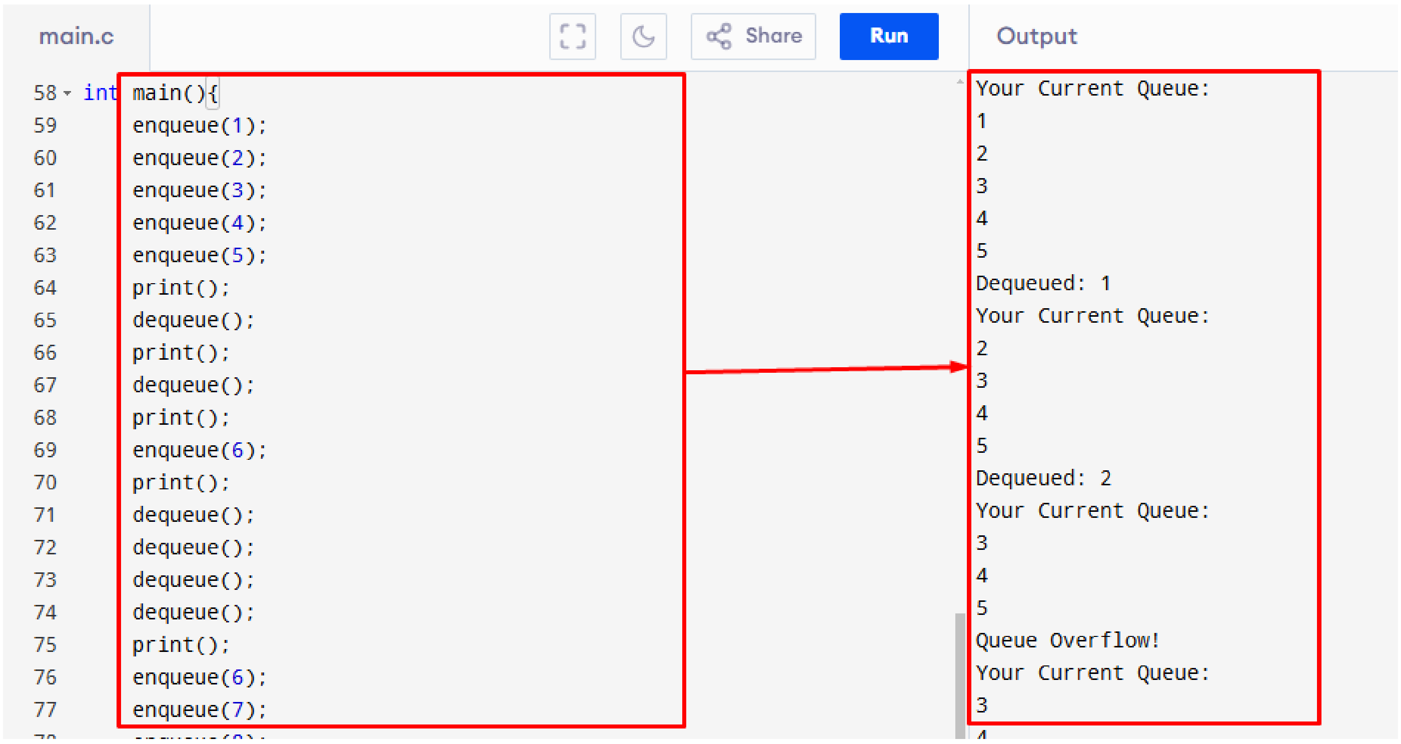
```
Before Swaping A = 20, B = 10.

After Swaping  A = 10, B = 20.


=== Code Execution Successful ===
```

Date: **26-11-2024**
Problem No: **19**
Problem Statement: **Implement a Queue to Support Efficient Creation, Data Insertion, and Deletion Operations.**

```
#include<stdio.h>
int front = -1;
int rear = -1;
int queue[5];
int isFull(){
   if (rear == 4){
      return 1;
   }else{
      return 0;
   } }
int isEmpty(){
   if (front == -1 || front > rear){
      return 1;
   }
```

```
    else{
        return 0;
    } }
void enqueue(int data){
    if (isFull()){
        printf("Queue Overflow!\n");
        return;
    }
    if (front == -1){
        front = 0;
    }
    rear = rear + 1;
    queue[rear] = data;
}
void dequeue(){
    if (isEmpty()) {
        printf("Queue is empty!\n");
        return;
    }
    printf("Dequeued: %d\n", queue[front]);
    front = front + 1;
}
void print(){
    if (isEmpty()){
        printf("Empty Queue\n");
        return;
    }
    printf("Your Current Queue:\n");
    for (int i = front; i <= rear; i++){
        printf("%d\n", queue[i]);
    }
}
```

```
int main(){
    enqueue(1);
    enqueue(2);
    enqueue(3);
    enqueue(4);
    enqueue(5);
    print();
    dequeue();
    print();
    dequeue();
    print();
    enqueue(6);
    print();
    dequeue();
    dequeue();
    dequeue();
    dequeue();
    print();
    enqueue(6);
    enqueue(7);
    enqueue(8);
    enqueue(9);
    enqueue(10);

    return 0;
}
```



```c
int main(){
    enqueue(1);
    enqueue(2);
    enqueue(3);
    enqueue(4);
    enqueue(5);
    print();
    dequeue();
    print();
    dequeue();
    print();
    enqueue(6);
    print();
    dequeue();
    dequeue();
    dequeue();
    dequeue();
    print();
    enqueue(6);
    enqueue(7);
```

```
Your Current Queue:
1
2
3
4
5
Dequeued: 1
Your Current Queue:
2
3
4
5
Dequeued: 2
Your Current Queue:
3
4
5
Queue Overflow!
Your Current Queue:
3
```

The End