Department of
Computer Science and Engineering

Title: Pointers and Arrays

Structure Programming Lab
CSE 104

Green University of Bangladesh

1 Introduction

In C when we define a pointer variable we do so by preceding its name with an asterisk. We also give our pointer a type which, in this case, refers to the type of data stored at the address we will be storing in our pointer. For example, consider the variable declaration:

$int * ptr;$

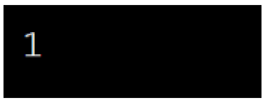
ptr is the name of our variable, the '*' informs the compiler that we want a pointer variable, i.e. to set aside however many bytes is required to store an address in memory. The int says that we intend to use our pointer variable to store the address of an integer.

The pointer can be used to access array elements. For example,

Code:

```c
#include<stdio.h>

int main()
{
  int arr[5] = { 1, 2, 3, 4, 5 };
  int *ptr = arr;

  printf("%d\n", *ptr);
  return 0;
}
```

Output:



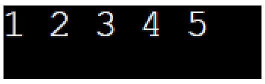
```
1
```

In this code, we have a pointer ptr that points to the 0^{th} element of the array. Similarly, we can point to whole array instead of only one element of the array.

Code:

```c
#include<stdio.h>

int main()
{
  int i, arr[5] = { 1, 2, 3, 4, 5 };
  int *ptr = arr;

  for(i=1;i<=5;i++)
  {
    printf("%d ", *ptr);
    ptr++;
  }

  return 0;
}
```

Output:



```
1 2 3 4 5
```

Example 1: Write a C program to find the sum of array elements using pointers.

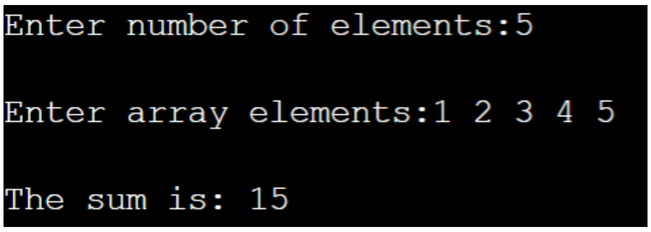
Algorithm 1 Steps in pseudo code:

```
1: Step 1: Start
2: Step 2: Declare array a[n] and variable i, n, sum, pointer *p
3: Step 3: Read n from user and read all the elements of a[n]
4: Step 4: Initialize variable i=0 and sum=0
5: Step 5: Assigned p=a;
6: Step 6: Repeat Until i<=n-1
7:     sum+=*p;
8:     i++, p++;
9: Step 7: Print "The sum of elements is": sum
10: step 8: end
```

Code:

```c
#include<stdio.h>
int main()
{
   int array[50];
   int i,sum=0;
   int *ptr;

   printf("\nEnter array elements:");
   for(i=0;i<5;i++)
      scanf("%d",&array[i]);

   /* array is equal to base address
    * array = &array[0] */
   ptr = array;

   for(i=0;i<5;i++)
   {
      //*ptr refers to the value at address
      sum = sum + *ptr;
      ptr++;
   }

   printf("\nThe sum is: %d",sum);
}
```

Output:



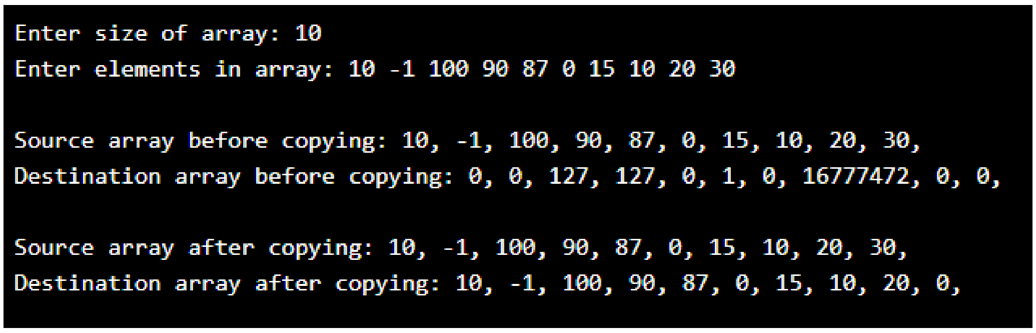
```
Enter number of elements:5

Enter array elements:1 2 3 4 5

The sum is: 15
```

Example 2: Write a C program to copy one array elements to another using pointers.

Code:

```c
#include <stdio.h>

#define MAX_SIZE 100 // Maximum array size


/* Function declaration to print array */
void printArray(int arr[], int size);


int main()
{
    int source_arr[MAX_SIZE], dest_arr[MAX_SIZE];
    int size, i;

    int *source_ptr = source_arr;   // Pointer to source_arr
    int *dest_ptr   = dest_arr;     // Pointer to dest_arr

    int *end_ptr;

    printf("Enter size of array: ");
    scanf("%d", &size);
    printf("Enter elements in array: ");
    for (i = 0; i < size; i++)
    {
        scanf("%d", (source_ptr + i));
    }

    // Pointer to last element of source_arr
    end_ptr = &source_arr[size - 1];

    /* Print source and destination array before copying */
    printf("\nSource array before copying: ");
    printArray(source_arr, size);

    printf("\nDestination array before copying: ");
    printArray(dest_arr, size);


    while(source_ptr <= end_ptr)
    {
        *dest_ptr = *source_ptr;

        // Increment source_ptr and dest_ptr
        source_ptr++;
        dest_ptr++;
    }


    /* Print source and destination array after copying */
    printf("\n\nSource array after copying: ");
    printArray(source_arr, size);

    printf("\nDestination array after copying: ");
    printArray(dest_arr, size);


    return 0;
}


void printArray(int *arr, int size)
{
    int i;

    for (i = 0; i < size; i++)
    {
        printf("%d, ", *(arr + i));
    }
}
```

Output:



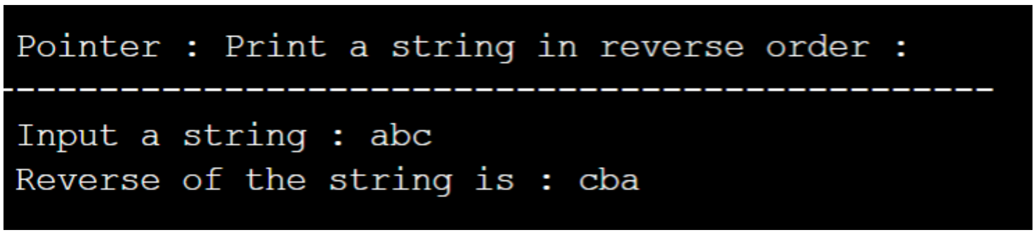
```
Enter size of array: 10
Enter elements in array: 10 -1 100 90 87 0 15 10 20 30

Source array before copying: 10, -1, 100, 90, 87, 0, 15, 10, 20, 30,
Destination array before copying: 0, 0, 127, 127, 0, 1, 0, 16777472, 0, 0,

Source array after copying: 10, -1, 100, 90, 87, 0, 15, 10, 20, 30,
Destination array after copying: 10, -1, 100, 90, 87, 0, 15, 10, 20, 0,
```

Example 3: Write a program in C to print a string in reverse using a pointer.

Code:

```c
#include <stdio.h>
int main()
{
    char str1[50];
    char revstr[50];
    char *stptr = str1;
    char *rvptr = revstr;
    int i=-1;
  printf("\n\n Pointer : Print a string in reverse order :\n");
  printf("------------------------------------------------\n");
    printf(" Input a string : ");
    scanf("%s",str1);
    while(*stptr)
    {
     stptr++;
     i++;
    }
    while(i>=0)
    {
     stptr--;
     *rvptr = *stptr;
     rvptr++;
     --i;
    }
    *rvptr='\0';
    printf(" Reverse of the string is : %s\n\n",revstr);
    return 0;
}
```

Output:



```
Pointer : Print a string in reverse order :
----------------------------------------------
Input a string : abc
Reverse of the string is : cba
```

2 Discussion & Conclusion

Based on the focused objective(s) to understand about the pointer operations, the additional lab exercise made me more confident towards the fulfilment of the objectives(s).

3 Lab Task (Please implement yourself and show the output to the instructor)

1. Write a program in C to find the maximum number between two numbers using a pointer.
2. Write a C program to swap two numbers using pointers.
3. Write a program in C to print all the alphabets using a pointer.
4. Write a C program to search an element in array using pointers.

4 Lab Exercise (Submit as a report)

- Write a program in C to find the maximum and minimum element in an array using pointer.
- Write a C program to convert Decimal to Binary number system pointer.
- Write a C program to concatenate two strings using pointers.
- Write a C program to add two matrix using pointers.

5 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.