The BRITISH UNIVERSITY IN EGYPT

Introduction to Computing

Lab (9)

Exercise 1: Write C++ program to copy the content of an array of characters to another array using pointers:

Solution (1)

```
#include <iostream>
using namespace std;
const int MAX_SIZE = 100; // Maximum size of the string
void main() {
    char text1[MAX_SIZE], text2[MAX_SIZE];
    char* str1 = text1;
    char* str2 = text2;

    // Inputting string from user
    cout << "Enter any string: " << endl;
    cin >> text1;

    // Coping text1 to text2 character by character
    while (*(str2++) = *(str1++));

    cout << "First string: " << text1 << endl;
    cout << "Second string: " << text2 << endl;
}
```

Exercise 2: Write C++ program to print all the alphabets using pointers:

Solution (2)

Exercise 3: Write a C++ program according to the following:

1. Import the iostream header file. This will allow us to use the functions defined in the header file without getting errors.
2. Include the std namespace to use its classes without calling it.
3. Call the main() function. The program logic should be added within the body of this function. The { marks the beginning of the function's body.
4. Declare an integer variable x and assigning it a value of 27.
5. Declare a pointer variable *ip.
6. Store the address of variable x in the pointer variable.
7. Print some text on the console.
8. Print the value of variable x on the screen.
9. Print some text on the console.
10. Print the address of variable x. The value of the address was stored in the variable ip.
11. Print some text on the console.
12. Print value of stored at the address of the pointer.
13. The program should return value upon successful execution.
14. End of the body of the main() function.

Solution (3)

Exercise 4: Write a C++ program according to the following:

1. Declare an integer pointer variable ip.
2. Declare an array named arr and store 6 integers into it.
3. Assign arr to ip. The ip and arr will become equivalent.
4. Create a for a loop. The loop variable x was created to iterate over the array elements from index 0 to 5.
5. Print the values stored at the address of the pointer IP. One value will be returned per iteration, and a total of 6 repetitions will be done. The endl is a C++ keyword that means the end line. This action allows you to moves the cursor to the next line after each value is printed. Each value will be printed in an individual line.
6. To move the pointer to the next int position after every iteration.
7. End of the for a loop.
8. The program must return something upon successful execution.
9. End of the main() function body.

Solution (4)

Exercise 5: Write a C++ program according to the following:

Note that: if there is no exact address that is to be assigned, then the pointer variable can be assigned a NULL. It should be done during the declaration. Such a pointer is known as a null pointer. Its value is zero and is defined in many standard libraries like iostream.

1. Declare a pointer variable ip and assigning it a value of NULL.
2. Print value of pointer variable ip alongside some text on the console.
3. The program must return value upon successful completion.
4. End of the body of the main() function.

Solution (5)

Exercise 6 – complete the following program: Write C++ program to find sum of integers in an array using pointers:

Solution (6)

```
#include <iostream>
using namespace std;

const int MAX_SIZE = 5;
void main() {
   int var[] = {10, 20, 30, 40, 50};
   int i = 0, sum = 0;
   int *pt;
_____
_____
_____
_____
_____
_____
_____
_____

   cout << "Sum of Array : " << sum;
}
```

Exercise 7: Write C++ program to define a function ***swap.*** This function swaps two numbers using pointers.

Solution (7)

```
#include <iostream>
using namespace std;

void swap(int* num1, int* num2) {
    int temp;
    //Copy the value of num1 to some temp variable
    temp = *num1;

    //Copy the value of num2 to num1
    *num1 = *num2;

    //Copy the value of num1 stored in temp to num2
    *num2 = temp;
}
void main() {
    int num1, num2;

    //Inputting 2 numbers from user
    cout << "Enter the first number : ";
    cin >> num1;
    cout << "Enter the Second number : ";
    cin >> num2;

    //Passing the addresses of num1 and num2
    swap(&num1, &num2);

    //Printing the swapped values of num1 and num2
    cout << "First number : " << num1<<endl;
    cout << "Second number: " << num2;
}
```

Exercise 8: Write a C++ program to input elements in array and search for an element entered by the user in this array using pointers.

1. Define a constant integer MAX_SIZE and set its value to 50.
2. Define an array and set its initial size to the value of MAX_SIZE.
3. Ask the user to enter the size of the array and its values. Take the values of the array through function, ***inputArray***, that takes as parameters the array and the size (entered by the user).
4. Ask the user to enter the element to search for.
5. Send this element with the array and its size to function, ***search***, to search the array for the element entered by the user. This function should return the index in which the element was found.

The prototype of the two functions should be as follows:

```
void inputArray(int * arr, int size);
int search(int * arr, int size, int toSearch);
```

Solution

Exercise 9: Write down the tracing of the following C++ program:

Problem (1)

```
#include <iostream>
using namespace std;
int x1 = 2;
void calc(int arr[], int g) {
  x1 = arr[g];
  arr[g] = arr[g + 1];
  arr[g + 1] = x1;
}
int change(int x2, int& x3) {
  return x1 + x2 + ++x3;
}
void main() {
  int x4[4] = { -1, 1, 2, x1 };
  int x5 = 1; x1 = 2;
  int x6 = change(*x4, x5);
  cout << x6 + x5 << " - ";
  calc(x4, 2);
  x6 = change(*x4, x5);
  cout << x6 + x5 << endl;
}
```

Solution

Problem (2)

```
#include <iostream>;
using namespace std;
struct student {   int mark = 0;
                   string name;
};
void main() {
   student students[3];
   (*students).mark = 1;
   (*students).name = "Ahmed";
   (*(students + 2)).mark = 5;
   (*(students + 1)).name = "Noha";
   (*(students + 2)).name = "Abbas";
   int min = 100;
   int max = 0;
   string BestStudent;
   string worstStudent;
   for (int i = 0; i < 3; i++) {
      if (min > students[i].mark) { worstStudent = students[i].name;
                                    min = students[i].mark; }
      if (max < students[i].mark) {  BestStudent = students[i].name;
                                     max = students[i].mark; }
   }
   cout << "worst Student is " << worstStudent << "\n";
   cout << "Best Student is " << BestStudent;
}
```

Solution