SOF103 C and C++ Programming
Lab Exercise 6 (week 7)
Pointers

Part A: Review Questions

1. Give two separate statements that assign the starting address of array `numbers` to the pointer variable `nPtr`.
2. Print the elements of array `numbers` using pointer/offset notation with pointer `nPtr`.
3. Print the elements of array `numbers` by subscripting pointer `nPtr`.
4. Assume that the variable `ptr` is of type **`char *`** and arrays `s1`[100] and `s2`[100] are of type **`char`**.
   a. Copy the string stored in array `s2` into array `s1`.
   b. Assign the address of the first element in `s2` to pointer `ptr`. Then, use `ptr` to print the characters of `s2`.
5. Consider the code below:

```
int c1 = 1, c2 = 2;
int  *p1 = &c1;
int *p2, *p3;
p2 = &c2;
cout << "*p2 = " << *p2 << endl;
p2 = p1;
cout << "*p2 = " << *p2
     << ", p2 = " << p2 << endl;
p3 = p2;
cout << "*p3 =" << *p3
     << ", p3 = " << p3 << endl;
```

   Assume address `c1` is 1000 and address `c2` is 2000, what is the output printed on the screen?

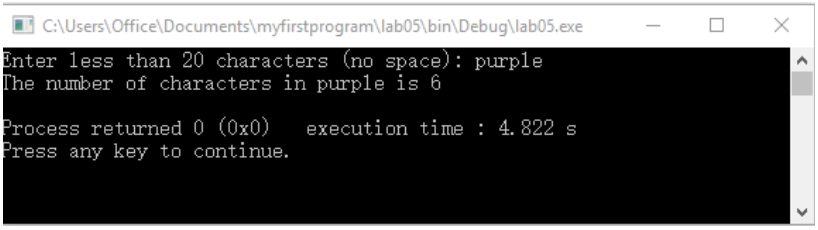
6. Assume that unsigned integers are stored in 2 bytes and starting address of array `values` is at location 2500. Answer the following questions related to array values.
   a. Declare an array of **`unsigned int`** called `values` with 5 elements, and initialize the elements to the even integers from 2 to 10. Assume the symbolic constant `SIZE` has been defined as 5.
   b. Declare a pointer `ptr` that points to an object of type **`unsigned int`**.
   c. Print the elements of array `values` using array subscript notation.
   d. Assign `ptr` to point to array `values`.
   e. Print the elements of array `values` by subscripting the pointer to the array.
   f. What address is referenced by `ptr + 3`? What value is stored at that location?
   g. Assuming `ptr` points to `values`[4], what address is referenced by `ptr -= 4`?

Part B: Programming Exercises

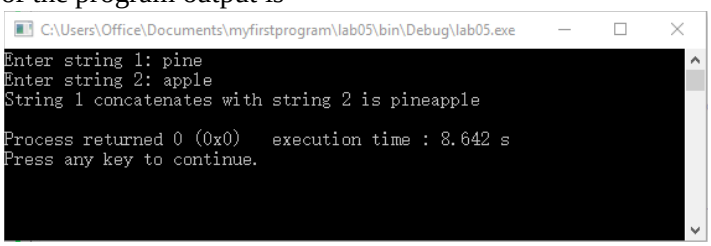
1. Write a program to accept 5 integer values from keyboard. The 5 values will be stored in an array using a pointer. Then print the elements of the array on the screen.
2. Modify the solution in exercise 1 above in order to print the elements of the array in reverse order using a pointer.
3. Write a program to accept 5 characters from user and then print out the characters in reverse order.
4. Run the following code below. Using `cout` statements to find the value of left-hand side variable in each assignment statements labelled from 1 to 12. Try to understand the statements and their outputs.

```
int main()
{
   char blocks[3] = {'A','B','C'};
   char *ptr = &blocks[0];
   char temp;

   temp = blocks[0];              // statement 1: temp = ?
   temp = *(blocks + 2);          // statement 2: temp = ?
   temp = *(ptr + 1);             // statement 3: temp = ?
   temp = *ptr;                   // statement 4: temp = ?

   ptr = blocks + 1;              // statement 5: *ptr = ?
   temp = *ptr;                   // statement 6: temp = ?
   temp = *(ptr + 1);             // statement 7: temp = ?

   ptr = blocks;                  // statement 8: *ptr = ?
   temp = *++ptr;                 // statement 9: temp = ?
   temp = ++*ptr;                 // statement 10: temp = ?
   temp = *ptr++;                 // statement 11: temp = ?
   temp = *ptr;                   // statement 12: temp = ?

   return 0;
}
```

5. Write a program to accept less than 20 characters from keyboard. Then, count the number of characters entered (not including the terminating character '\0'). Print the output on screen. An example of program output is shown below:



6. Write a program to prompt the user to enter two strings, then attach the second string to the end of the first string to form a new string. This process is called *string concatenation*. Store the two strings in separate character arrays with the suitable size. An example of the program output is

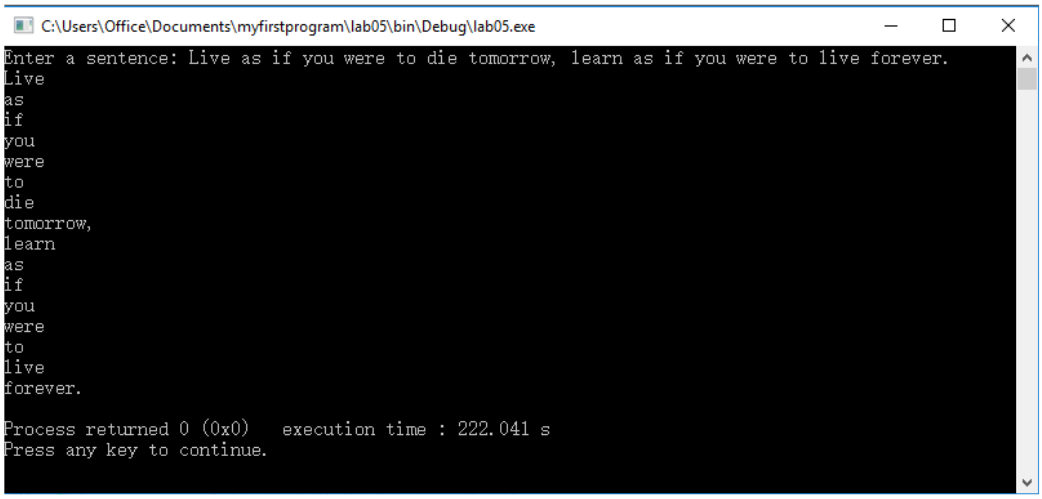


7. The function `getline()` is used to receive a string i.e. a line of characters (including white spaces) from user input. For example,

```
cin.getline(string1, 50);
```

obtains user input and store in variable `string1`, limited to the first 50 characters.

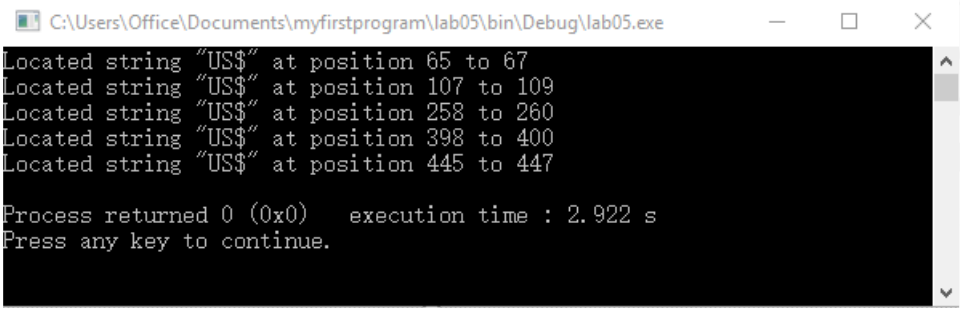
Write a program that accepts a sentence type in by the user using `getline()`, and then print out the individual words in the sentence. An example is shown below.



8. The following text is extracted from an online news article. (Reference: https://www.nst.com.my/business/2018/06/381187/bilateral-trade-between-malaysia-china-exceeds-us100-bln)

> Bilateral trade between Malaysia and China is expected to exceed US$100 billion this year, higher than the US$96 billion chalked up in 2017, said China's ambassador to Malaysia, Bai Tian. He said bilateral trade between the two nations grew 14.7 percent to US$42.7 billion in the first five months of this year compared with the same period last year. Imports from China to Malaysia accounted for US$18 billion, while exports to China stood at US$25 billion.

Write a program to search and locate the characters "US$" from the news article. Print out the location of the characters "US$". An example is shown below:



9. Modify your code in exercise 8 by replacing the characters "US$" appearing in the text with characters "MYR". Print out the text after replacement.