Midterm of CS 135, v1
Oct 17, 2024

Name:	
EmpID:	

1 (30 points) Answer the following questions.

(1) Given `char arr[] = {'A', 'B', 'C'}`, what is arr[1]?

Answer: arr[1] is `'B'`, the second item of array `arr`.

(2) Declare function **increase**, given an integer array `arr` with `size` many elements, increase each element of the array by 1. Return type is void. Define the function header (no implementation is needed).

Answer: `void increase(int arr[], int size);` or `void increase(int* arr, int size);`

Warning: `void increase(int& arr[], int size);` is wrong, need to replace `int& arr[]` by `int arr[]`.

(3) Assume that n is properly declared and initialized. Write a statement to declare `lastDigit` as an integer and initialize it to be the least significant digit of integer n. Suppose n is 123, after the statement, `lastDigit` is 3.

Answer: `int lastDigit = n % 10;`

(4) What is the output?

```
1  string tens_name(int n);
2
3  int main() {
4      cout << tens_name(82) << endl;
5      return 0;
6  }
7
8  string tens_name(int n) {
9      if (n < 20 || n > 99)
10        return "";
11
12     string names[] = {"", "", "twenty", "thirty", "forty",
13         "fifty", "sixty", "seventy", "eighty", "ninety"};
14
15     return names[n / 10];
16 }
```

Answer: eighty

(5) Given `string greeting = "How are you?";` What is the value for greeting.substr(4, 5)?

Answer: substring "are y"

(6) What is the value of 2 - 3 / 2?

Answer: 1

(7) The area of a trapezoid with bases a, b, and height h is $\frac{a+b}{2}h$. Assume that a, b, h are properly declared as double types and initialized, write a statement to declare *area* and save the value of the area.

Answer: `double area = 1 / 2.0 * (a+b) * h;` or `double area = (a + b) / 2 * h;`

(8) What is the output of the following code?

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

int main() {
    int count = 0;
    for (int i = 9; i >= 2; i -= 3)
        count++;

    cout << count << endl;
    return 0;
}
```

Answer: 3

(9) Write a statement to call `foo` function on integer variables a and b, both are properly declared and initialized.

```
void foo(int& a, int& b);
```

Answer: foo(a, b);

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

void foo(int& a, int& b);

int main() {
    int a = 3;
    int b = 4;
    foo(a, b);
    cout << "a = " << a << ", b = " << b << endl;
```

```
    return 0;
}

void foo(int& a, int& b) {
   int temp = a;
   a = b;
   b = temp;
}
```

(10) Write a condition in C++ to represent that an integer variable n is in the range of [60, 80], where both ends are included, that is, $60 \leq n \leq 80$ in mathematic representation.

Answer: solution 1: `n >= 60 && n <= 80`

solution 2: `n >= 60 and n <= 80`

2 (20 points) Answer the following questions.

(1) What is the output of foo(3, 4)?

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

void foo(int width, int height) {
    int mid;
    if (height % 2 != 0)
       mid = height / 2;
    else mid = height / 2 - 1;

    for (int row = 0; row < height; row++) {
        for (int col = 0; col < width; col++) {
            if (height % 2 != 0) {
               if (row == mid)
                  cout << "*";
               else cout << "-";
            }
            else //now height % 2 == 0
            {
               if (row == mid || row == mid + 1)
                  cout << "*";
               else cout << "-";
            }
        }

        cout << endl;
    }
```

```
}
```

Answer:

```
---
***
***
---
```

(2) Define function `numBigLetters`, for a string, return the number of characters that are capital letters, that is, letter from 'A' to 'Z'. **No need to include libraries.**

Hint: you may use `int isupper(int ch)` to test whether a character is uppercase letter or not.

Define main function with the following requests.

- Enter two strings from console. The strings may contain spaces.
- If both strings have the same number of uppercase letters, report "the strings have the same number of uppercase letters.", otherwise, find out and print the string with more uppercase letters. Some sample outputs are as follows.

```
Enter the first string: abc A
Enter the second string: bcd BB
bcd BB has more uppercase letters

Enter the first string: abcAB
Enter the second string: cd CD
the strings have the same number of uppercase letters
```

Answer:

```
//Sample input/output:
//Enter the first string: abc A
//Enter the second string: bcd BB
//bcd BB has more uppercase letters

//Enter the first string: abcAB
//Enter the second string: cd CD
//the strings have the same number of uppercase letters

#include <iostream>
using namespace std;

int numBigLetters(string str);

int main() {
```

```
    cout << "Enter the first string: ";
    string s1;
    getline(cin, s1);

    cout << "Enter the second string: ";
    string s2;
    getline(cin, s2);

    int n1 = numBigLetters(s1);
    int n2 = numBigLetters(s2);

    if (n1 > n2)
       cout << s1 << " has more uppercase letters" << endl;
    else if (n2 > n1)
           cout << s2 << " has more uppercase letters" << endl;
    else cout << "the strings have the same number of uppercase letters" << endl;

    return 0;
}

int numBigLetters(string str) {
   int count = 0;
   for (int i = 0; i < str.length(); i++)
       if (str[i] >= 'A' && str[i] <= 'Z')
          count++;

   return count;
}
```

3 (50 points) Programming exercises

(1) Define function called `sumEvenFactors`, for a positive integer, sum up its *non − trivial* factors that are even. A non-trivial factor of n is a factor of n other than 1 and itself.

Hint: $n/2$ may be a non-trivial factor of n.

Answer:

One implementation is as follows.

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

int sumEvenFactors(int n);

int main() {
```

```
    cout << sumEvenFactors(5) << endl; //0
    cout << sumEvenFactors(8) << endl; //6
    return 0;
}

int sumEvenFactors(int n) {
   int sum = 0;

   //Let i be even integer only
   for (int i = 2; i <= n/2; i += 2) {
       if (n % i == 0) //i is a factor of n
          sum += i;
   }

   return sum;
}
```

Here is another implementation.

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

int sumEvenFactors(int n);

int main() {
    cout << sumEvenFactors(5) << endl; //0
    cout << sumEvenFactors(8) << endl; //6
    return 0;
}

int sumEvenFactors(int n) {
   int sum = 0;

   for (int i = 2; i <= n/2; i++) {
       //Approach 1:
       //if (n % i == 0)
       //   if (i % 2 == 0)
       //      sum += i;

       //Approach 2:
       if (n % i == 0 && i % 2 == 0)
          sum += i;
   }

   return sum;
}
```

In main function, call function on integer 8. Print out the return. **Just write the statements in main function, no need to include libraries**.

Answer:

```
cout << sumEvenFactors(8) << endl; //6
```

(2) Write code in main to enter a full name in the format "FirstName LastName" (without quotes), extract the first name and last name and get the initial. **No need to include libraries**.

Here is a sample input/output, input is highlighted:

```
Enter full name in the format of firstName lastName: George Washington
Initial for George Washington is GW
```

Hints:

- Find out the index of the character separating first name and last name.

  `size_t find (char c, size_t pos = 0) const;`

  Searches the string for the first occurrence of character `c`. If you do not specify parameter `pos`, then the search starts from the beginning of the string. `size_t` is non-negative integer.
- Extract first name and last name.

  `string substr (size_t pos = 0, size_t len = npos) const;`

  Generate substring that is the portion of the object that starts at character position `pos` and spans `len` characters (or until the end of the string, whichever comes first). If the second parameter `len` is not provided, return a substring starting from `pos` all the way to last character.
- Initialize the result to be an empty string.
- Use concatenate operator + to add the first letter of first name to the result.
- Use concatenate operator + to add the first letter of last name to the result.

Answer:

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

int main() {
    cout << "Enter full name in the format of firstName lastName: ";
    string fullName;
    getline(cin, fullName);

    int spIdx = fullName.find(' ');
    string firstName = fullName.substr(0, spIdx - 1);
    string lastName = fullName.substr(spIdx + 1);

```

```
    //warning: cannot write string initial = firstName[0] + lastName[0]; or
    //string initial = "" + firstName[0] + lastName[0];
    string initial = "";
    initial += firstName[0];
    initial += lastName[0];
    cout << "Initial for " << fullName << " is " << initial << endl;

    return 0;
}
```

(3) Define function `longerThan`, given an array of strings, and its size, together with a target string, if every element of the array has more characters than that of the target, then return true. If there is at least one element of the array has same number as or fewer number of characters than that of the target, return false.

Hint: `length` or `size` method of string class returns the number of characters of that string.

Answer:

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

bool longerThan(string arr[], int size, string target);

int main() {
    string arr[] = {"abc", "bcef", "bcb"};
    int size = sizeof(arr) / sizeof(arr[0]);

    cout << boolalpha << longerThan(arr, size, "bc") << endl; //true
    cout << boolalpha << longerThan(arr, size, "abc") << endl; //false

    return 0;
}

bool longerThan(string arr[], int size, string target) {
    for (int i = 0; i < size; i++)
        if (arr[i].length() <= target.length())
           //length can be replaced by size
           return false;

    return true;
}
```

In main function, write the following statements.

- Declare an array of strings, call it `arr`, initialized with elements `"abc", "bcef", "bcb"`.

- Call the above function on `arr` and target `"bc"`, print out the result.

Answer:

```
string arr[] = {"abc", "bcef", "bcb"};
int size = sizeof(arr) / sizeof(arr[0]);

cout << boolalpha << longerThan(arr, size, "bc") << endl; //true
```