CS111 Lab 26

Goal: Learn how to use string functions. Practice on thinking string as character array.

Before you start working on the exercises below, go through the example codes and reference materials in today's lab resource section to find out how the following functions work:

strlen(str), str.size(), str.length()	str.find(str1), str.rfind(str1)
str.substr(pos, length)	str.insert(pos, str2)

1) Consider the following C++ program and write an expected output for lines a-b.

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

int main() {
      string words[4] = {"How","are","you?",""};
      for (int i = 1; i <= 2; i++) cout << words[i];   //line a)
      cout << endl;
      for (int i = 0; i <= 2; i++) cout << words[i][i];//line b)
      cout << endl;
      return 0;
}
```

Answer:

a) areyou? b)Hru

2) Consider the following C++ program:

```
int main() {
      string s = "Hello";
      // (a)
      // (b)
      return 0;
}
```

(a) Write a statement (hint: for loop) that prints the string s in reverse (i.e. will output olleH):

```
for (int i = 4; i >= 0; i--) cout << s[i];
```

(b) Write a statement that prints the length of the string s (apply appropriate function!):

```
cout << s.length();
```

Title Lines:

3) (prac2.pdf and prac3.pdf) Write **title lines** for the functions that are called by the following main program. **Do not supply the blocks for the functions.**

```
int main() {
   char c = 'A'; int b = 1961;
   cout << numSixes("19683") << endl;       // (a) prints 1
   printNumSixes(19683);                    // (b)  prints 1
   cout << longest(b, b, 5) << endl;        // (c) prints 1961
   average(2.5, 3.4, 4.0);                  // (d) prints 3.3
   c = randomLetter(); cout << c << endl; // (e) prints random letter eg Z
   return 0;
}
```

(a) Title line for **numSixes** `int numSixes(string a)`

(b) Title line for **printNumSixes** `void printNumSixes(int x)`

(c) Title line for **longest** `int longest(int a, int b, int c)`

(d) Title line for **average**

`void average(double a, double b, double c)`

(e) Title line for **randomLetter** `char randomLetter()`

4) (prac3.pdf) Write title lines (header lines or prototypes) for the following functions. Do not supply the blocks for the functions.

```
int main() {
   int b[5] = {9, 3, 0, 4, 7};
   int x = 17;
   cout << integerPart(3.14159) << endl; // (a) prints 3
   swap1(x, b[1]);                       // (b) swaps b[1] with x
   swap2(b, 1, x);                       // (c) swaps b[1] with x
   median(x +1, x, x+2);                 // (d) prints 18 the median value
   cout << sqrt(5, 10, 12) << endl;
   // (e) prints "Error" for any input values
   return 0;
}
```

(a) Title line for **integerPart** as called at the line marked (a).

`int integerPart(double x)`

(b) Title line for **swap1** as called at the line marked (b).

`void swap1(int &a, int &b)`

(c) Title line for **swap2** as called at the line marked (c).

`void swap2(int a[], int i, int &b)`

(d) Title line for **median** as called at the line marked (d).

`void median(int a, int b, int c)`

(e) Title line for **sqrt** as called at the line marked (e).

`string sqrt(int a, int b, int c)`

Tracing through a program for output:

5) (prac3.pdf) Consider the following C++ program.

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

string fun(int x) {
     string ans = "0123456789";
     if (x <= 0) return "4";
     if ((x >= 30) && (x < 1000)) return ans.substr(x % 7);
     if ((x >= 0) || (x < 100)) return "x11";
     return ans;
}

int up(int &x) {
     x--;
     cout << x << endl;
     return x - 1;
}

int main() {
     int x = 5;
     cout << fun(0) << endl;        // line (a)
     cout << fun(33) << endl;       // line (b)
     cout << fun(3003) << endl;     // line (c)
     up(x);                         // line (d)
     cout << up(x) << endl;         // line (e)
     return 0;
}
```

(a) What is the output at line (a)? 4

(b) What is the output at line (b)? 56789

(c) What is the output at line (c)? x11

(d) What is the output at line (d)? 4

(e) What is the output at line (e)? 3
2

6) (prac3.pdf) Consider the following C++ program.

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

string fun(int x) {
      if (x <= 0) return "";
      if (x >= 9 && x % 2 == 1) return "x+1";
      if (x >= 9 || x % 3 == 0) return "x+2";
      return "5";
}

int rec(int x) {
      if (x < 100) return x/5;
      return rec(x / 10) + rec(x % 100);
}

int main() {
      cout << fun(-3) << endl;   // line (a)
      cout << fun(33) << endl;   // line (b)
      cout << rec(36) << endl;   // line (c)
      cout << rec(-555) << endl; // line (d)
      cout << rec(987) << endl;  // line (e)
      return 0;
}
```

(a) What is the output at line (a)?

(b) What is the output at line (b)? x+1

(c) What is the output at line (c)? 7

(d) What is the output at line (d)? -111

(e) What is the output at line (e)? 36

Short Blocks of code:

7) (prac3.pdf) Write blocks of code to perform the functions used in the following main program. Your blocks must match the given title lines. Each block should be a short function of only a few lines.

```
int main() {
      string s = "HELLO", t = "GOODBYE";
// (a) Do two strings have the same number of characters?
      cout << sameLength(s, t) << endl;
// (b) Tests whether a string contains a target
      cout << contains("HELL", s) << endl;
// (c) Returns a string that drops the last character
      cout << dropLast(t) << endl;
// (d) Prints the third character
      cout << third(t) << endl;
// (e) Turns an upper case character to lower case
      lower(s[0]);
      cout << s << endl;
      return 0;
}
```

(a)

```
bool sameLength(string x, string y) {
   return x.length() == y.length();
}
```

(b)

```
bool contains(string target, string x) {
   return ((int) x.find(target)) >= 0;
}
```

(c)

```
string dropLast(string x) {
   return x.substr(0, x.length() - 1);
}
```

(d)

```
char third(string x) {
   return x[2];
}
```

(e)

```
void lower(char &x) {
   if ('A' <= x && x <= 'Z') x = x + 'a' - 'A';
}
```

Writing a program or a function:

8) (prac3.pdf) Write a complete C++ program that does the following. (Programs that correctly carry out some of the tasks will receive partial credit.)

1. It asks the user to enter an integer n that is between 1 and 20.
2. It repeatedly reads n from the user until the supplied value of n is legal.
3. It prints out a square picture (as shown in the diagram, but with n rows) that uses the uppercase letters O and X in sequence, to form an outer perimeter of Os that contains a perimeter of Xs, that contains a permimeter of Os,

and so on.

Here is an example of how the program should work:

```
Give me an integer between 1 and 20: 7
OOOOOOO
OXXXXXO
OXOOOXO
OXOXOXO
OXOOOXO
OXXXXXO
OOOOOOO
```

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

int main() {
      char picture[20][20];
      int n = 0;

      while (n < 1 || n > 20) {
            cout << "Give me an integer between 1 and 20: ";
            cin >> n;
      }
      int mid = (n + 1) / 2;
      for (int perim = 0; perim < mid; perim++) {
            char x = 'O';
            if (perim % 2 == 1) x = 'X';
            for (int r = perim; r < n -perim; r++)
                  for (int c = perim; c < n -perim; c++)
                        picture[r][c] = x;
      }
      for (int r = 0; r < n; r++) {
            for (int c = 0; c < n; c++)
                  cout << picture[r][c];
            cout << endl;
      }
      return 0;
}
```

9) (prac3.pdf) Write a function called numX that reports the number of elements in a array of strings that contain an uppercase letter X.

For example, a program that uses the function follows.

```
int main() {
   string data[4] = {"abcdXYZ", "Hello", "1234", "XXX"};
   cout << numX(data, 4); // prints: 2 because 2 strings include an X
   return 0;
}
```

```
int numX(string a[], int cap) {
   int ans = 0;
   for (int i = 0; i < cap; i++)
      if (((int) a[i].find("X")) >= 0) ans++;
   return ans;
}
```