
COMP 2011 Midterm - Spring 2015 - HKUST

- Date: March 30, 2015 (Monday)
Time Allowed: 2 hours, 7:00 pm - 9:00 pm
Instructions:
1. This is a closed-book examination.
 2. There are **8** questions on **11** pages(including the cover page).
 3. Write your answers in the space provided in black/blue ink. **NO** pencil please.
 4. All programming codes in your answers must be written in **ANSI C++**.
 5. You may use **only** the C++ language features and constructs learned in class so far. For example, no pointers, C++ classes, string class, etc.
 6. For programming questions, you are **NOT** allowed to define additional helper functions or structures, nor global variables unless otherwise stated. you also **can not** use any library functions not mentioned in the questions.

Student Name	
Student ID	
Email Address	

Problem	Score
1	/30
2	/8
3	/8
4	/8
5	/10
6	/10
7	/10
8	/16

Problem 1 [30 points] True or False

Indicate whether the following statements are *true or false* by circling T or F. You get 2 points for each correct answer, -1 for each wrong answer, and 0 if you do not answer.

T F (1) **while** is a program control instruction. **T**

T F (2) "_something2" is not a legal variable name. **F**

T F (3) Numeric data type **double** has the larger range than numerical data type **integer**. **T**

T F (4) In the following code, the **cout** statement will execute 5 times. **F**

```
for (i = 0; i < 5; i++);  
cout << i;
```

T F (5) When **break** statement is encountered in **switch** statement, it stops the execution of the entire program. **F**

T F (6) The **continue** statement is used to terminate loops or to exit from a switch. **F**

T F (7) Any **if-else** statement can be replaced by **switch** statement. **F**

T F (8) C++ checks if an array index is out of bound while a program is running. **F**

T F (9) What is the truth value of the following boolean expression when D is 10, P is 2, and Q is 3. **T**

$(P < D/Q) \ \&\& \ (!P + Q < D)$

T F (10) The main body of program to calculate the n-th Fibonacci sequence is

```
int fib(int n) {  
    if (n==0) return 1;  
    else if (n==1) return 2;  
    else return fib(n-1)+fib(n-2);  
}
```

If we run *fib*(2), then the function *fib* will be executed twice. **F**

T F (11) If we want to express the if-statement condition " $10 < x < 100$ ", we can write it as **F**

```
if (10 < x < 100)
```

T F (12) The output of the following program is -1. **T**

```
#include<iostream>  
using namespace std;  
void main()  
{  
    int x=2;  
    while(x--);  
    cout<<x<<endl;  
}
```

T F (13) Suppose i, j and k are all integer variables, after executing the following for loop, k will become 9. **F**

```
for (i=0, j=10; i<=j; i++, j--)  
    k = i+j;
```

T F (14) >, <, = are all valid comparison operators in C++ language. **F**

T F (15) The output of the following program is 10 4. **T**

```
int x = 10, y = 3;  
cout<<x--<<' ' << ++y<<endl;
```

Problem 2 [8 points] Switch and Break

```
#include<iostream>
using namespace std;
void main(){
    int x = 1, y = 0, a = 0, b = 0;
    switch(x)
    {
        case 1: switch(y) {
                    case 0: a++; break;
                    case 1: b++; break;
                }
        case 2:
            a++;b++;break;
        case 3:
            a++;b++;
    }
    cout<<"a="<<a<<' \t ' <<"b="<<b<<endl;
}
```

The output of the program is _____

Answer: _____ a=2 b=1

Marking Scheme:

If you write a = 2, you get 3 points;

If you write b = 1, you get another 3 points;

If you get all the format right, you get another 2 points.

Problem 3 [8 points] Approximate Pi

A famous math formula to calculate pi is

$$\frac{\pi^2}{6} \approx \frac{1}{1^2} + \frac{1}{2^2} + \dots + \frac{1}{n^2} \quad (1)$$

The following program uses the formula to calculate pi, and the stopping criterion is satisfied when the newest item is $\frac{1}{n^2}$ less than 10^{-6} , which is indicated by the stopping criterion $i*i < 10e+6$ in the program below. Please fill in the blanks to complete the program.

```
#include<iostream>
using namespace std;
void main() {
    int i = 1;
    _____ pi = 0;
    while (i*i<10e+6)
    {
        pi = _____;
        i++;
    }
    pi = sqrt(6.0*pi);
    cout<<"pi ="<<pi<<endl;
}
```

The first line: _____

The second line: _____

Answer:

The first line: _____ double / float/ long double

The second line: _____ pi+1.0/(i*i)

Marking Scheme:

first line : 3 points;

second line: pi+ (2 points);1.0/(i*i) (3 points).

Problem 4 [8 points] Break and Continue

```
#include<iostream>
using namespace std;
void main()
{
    int i;
    for(i=1; i<=6;i++){
        if(i%3)
            cout<<"*";

        else
            continue;
        // "continue" is changed to "break" in part (b)
        cout<<"#";
    }
    cout<<"$";
}
```

(a). What is the output of the above program?

(b). If we replace the continue statement by a break statement, what is the new output?

Answer:

(a). What is the output of the above program?

*****\$

(b). If we replace the continue statement by break, what is the new output?

***\$

Marking Scheme:

question (a): 4 points in total. If you write an output end with \$, you get 1 point, if you write *****\$, you get 2 points.

question (b):4 marks in total. If you write an output end with \$, you get 1 point, if you write ***\$, you get 2 points.

Problem 5 [10 points] Parameter Parsing Methods

```
#include<iostream>
using namespace std;
void dd(int &x, int &y, int z)
{
    x = x + z;
    y = y - x;
    z = 10;
    cout<<x<<'\\t'<<y<<'\\t'<<z<<endl;
}

void main()
{
    int a = 3, b = 4, c = 5;
    for (int i = 0; i < 2; i++)
        dd(a, b, c);
    cout<<a<<'\\t'<<b<<'\\t'<<c<<endl;
}
```

The output of the program is _____

```
8   -4  10
13  -17 10
13  -17  5
```

Answer: The output is _____

Marking Scheme:

There are total nine numbers in the right answer, you earn 1 point for each right number, and you get 10 points if you get all the nine numbers right.

Problem 6 [10 points] Count the Words in a Sentence

An English sentence is read into a 1-dimensional C++ char array *s*. Suppose the sentence consists of only words or numbers separated by one or more spaces, and each word is comprised of the 26 characters in either lower or upper case. Implement the function *word_counting* to calculate the number of words(exclude the numbers in a sentence) in a sentence. Some examples are listed in the table below:

EXAMPLE	SENTENCE	OUTPUT
Example #1	"I have 3 apples"	3
Example #2	"China is a big country"	5
Example #3	"Lily and Jason are good friends "	6

Your function has one input: the sentence, *i* and returns the number of words in the given sentence.

Your function also has a local bool variable "flag", which you can use to identify a word. Note that the "flag" variable is very important in the function. At the beginning, the "flag" variable is set to be 0. When it encounters a character either in lower or upper case, which indicates the beginning of a valid word, it will change to 1. And at the end of a word, that is, when it encounters a space, the "flag" variable will again change from 1 to 0;

```
int word_counting(const char s[] )  
{  
    bool flag = 0;
```

```
}
```


Answer:

```
int word_counting(const char s[], bool flag)
{
    int i = 0, num = 0;
    flag=0;
    while(s[i]!='\0')
    {
        if((s[i]>='a'&& s[i]<='z' || s[i]>='A'&& s[i]<='Z') &&
            flag==0)
        {
            flag = 1;
            num++;
        }
        else if(s[i]==' ')
            flag = 0;
        i++;
    }
    return num;
}
```

Marking Scheme:

1. If you get the stopping criterion "if (s[i]!='\0') or similar expressions, you earn 2 points.
2. If you get the criterion for checking whether a character with either lower cases or upper cases like: *if((s[i] >= 'a' && s[i] <= 'z') || (s[i] >= 'A' && s[i] <= 'Z'))*, you get 3 points.
3. If you have a return sentence, you get another 2 points.
4. If you get all the other things correct, you will get as a total 10 points.

Problem 7 [10 points] Recursion

Gluttonous Tom is trying to sell his peaches. On the first day he sold half of his peaches and ate one. On the second day he sold half of his remaining peaches and ate one again, so on and so forth. On the beginning of the n th day, he found that he had only one peach left. Write a recursive function to compute how many peaches Tom originally has.

Sample Input: $(1 \leq n \leq 20)$

3

Sample Output:

10

```
int peach(int n)
{
}
```

Answer:

```
int peach(int n)
{
    if (n == 1)
    {
        return 1;
    }
    else
    {
        return (2 * (Peach(n - 1) + 1));
    }
}
```

Marking Scheme:

Judging whether n is between 1 and 20 is not necessary. If students do so, it is counted as correct.

1. The structure of the program is basically in a recursive form. (3 points)
2. The $n = 1$ case is 1 points. (1 point)
3. The recursive case is 6 points. (6 points)

Problem 8 [16 points] Substring Matching

Complete the following program to judge whether a is a substring of b . If 'yes', how many times does a appear in b ? For example,

1. $a = "aba"$, $b = "ababab"$, then a has appeared in b for 2 times.

2. $a = "aaa"$, $b = "aaaaaa"$, then a has appeared in b for 4, not 2 times.

We have implemented the program structure for you. You need to complete the missing part. The length of a and b is between 1 to 100.

```
#include <iostream>
#include <cstring>

using namespace std;

int substring(char a[], char b[], int la, int lb)
{
    // complete the function
}

int main()
{
    char a[100], b[100];
    cin >> a, cin >> b;

    // strlen() compute the length of a char array
    int la = strlen(a), lb = strlen(b);

    int count = substring(a, b, la, lb);
    if(count == 0)
    {
        cout << "a is not a substring of b.\n";
    }
    else
    {
        cout << "a has appeared in b for " << count << "
            times.\n";
    }
    return 0;
}
```

Answer:

```
int substring(char a[], char b[], int la, int lb)
{
    int count = 0;
    for(int i = 0; i <= lb - la; i++)
    {
        bool flag = true;
        for(int j = 0; j < la; j++)
        {
            if(a[j] != b[i+j])
            {
                flag = 0;
                break;
            }
        }
        if(flag)
        {
            count++;
        }
    }
    return count;
}

int substring(char a[], char b[], int la, int lb)
{
    // complete the function
    bool flag = 1;
    if(la > lb)
    {
        return 0;
    }
    for(int i = 0; i < la; i++)
    {
        if(b[lb-i-1] != a[la -i -1])
        {
            flag = 0;
            break;
        }
    }
    if(flag)
    {
        return 1+ substring(a, b, la, lb-1);
    }
    else
        return substring(a, b, la, lb-1);
};

int substring(char a[], char b[], int la, int lb)
{
    // complete the function
    int count = 0;
    for(int i = 0; i <= lb-la; i++)
    {
```

```

        for(int j = 0; j < la; j++)
        {
            if(a[j] != b[i+j])
            {
                count--;
                break;
            }
        }
        count++;
    }
    return count;
};

```

Marking Scheme:

1. If you write down the outer loop: $for(int i = 0; i \leq lb - la; i++)$, you can get 3 points. Some students write the stopping criterion as $i < lb - la$, this can get 2 points. (3 points)
2. The inner loop counts 3 points. (3 points)
3. You should have a counter variable to save the number of times a appears in b. (1 point)
4. The if statement $if(a[j] \neq b[i+j])$ (or equivalently $a[j] == b[i+j]$) is 2 points. (2 points)
5. The flag variable is 3 points. Some students may use other ways to judge the substrings. This is also correct as long as the answer is right. (3 points)
6. The break statement is 2 points. (2 points)
7. If the program is generally correct, you can get the remaining 2 points. (2 points)