CS2310 2025/26 Semester A Mid-Term Mock Test

1. Total time allowed: 120 minutes

2. Five questions and do not include any extra libraries, except <iostream> and <iomanip>

Name: Student ID: Seat number:

Question 1 [24 marks]

Write the console output of the following programs.

A.

```
int main() {
    int x = 5;
    int a, b;
    a = x++;
    b = ++a;
    cout << b << ',' << x;
    return 0;
}
```

ANS:

B.

```
char x = 'b'; // x+1='c'
char y = 'd';
char z = ++y;
cout << z - x;
```

ANS:

C.

```
int func(int a, int b) {
    int res = 0;
    for (int i = 0; i < a; i++) {
        if (i / 2 == 0) res += b;
    }
    return res;
}
int main() {
    int a = 4, b = 1;
    cout << func(a, b);
    return 0;
}
```

ANS:

D.

```
int func(char flag) {
```

```
    int result = 0;
    switch (flag) {
    case 'a':
        result = 3;
        break;
    case 'b':
        result = 2;
        break;
    case 'c':
        result = 1;
        break;
    default:
        result = 4;
    }
    return result;
}

int main() {
    int result = func('b') + func('c');
    cout << result;
    return 0;
}
```

ANS:

E.

```
char a[5] = {'a','b','c','d','e'};
for (int i = 4; i > 0; i--) {
    if (a[i] > 'c') continue;
    else if (a[i] == 'b') break;
    else
        cout << a[i];
}
```

ANS:

F.

```
int x = 3;
int y = x++;
if (x > y) cout << x + 1;
else cout << y - 1;
return 0;
```

ANS:

Question 2 [18 marks]

The function of the following program fragment is to calculate the number of odd numbers in an array and the average of these odd numbers. Check this program fragment for syntax errors and correct them. Marks will be deducted for incorrect answers.

Line	Given Program Code	**Correct Code (ANS)**
1	#including<iostream>	
2		
3	using namespace std;	
4	int main	
5	{	
6	int i, count, sum;	
7	int average;	
8	int a[] = { 1, 2, 3, 4, 5, 6, 7, 8};	
9	for(i = 0; i < =8; i++)	
10	{	
11	if(a[i] % 2 != 0)	
12	continue;	
13	sum = a[i]+sum;	
14	count+;	
15	}	
16	average == sum/count;	
17	cout >> "count = " >> count >> endl >> "average = " >> average >> endl;	
18	return 0	
19	}	

Question 3 [18 marks]

The formula f has two inputs: ***n*** (date type: **int**) and ***x*** (date type: **double**). The result of f is also **double** type.

$$f = x^1 + x^2 + x^3 + \cdots + x^n$$

Write a program according to the following requirements:

- For the inputs of ***n*** and ***x***, we assume that they are already the **int-type** and **double-type** numbers, respectively, i.e., you do not need to check whether they are belonged to any other data types.
- For the input ***x***, check whether $0.1 \leq x \leq 1$. If no, input ***x*** again.
- Next, for the input ***n***, check whether it is **positive** and **odd**. If no, input ***n*** again.
- If both ***n*** and ***x*** are valid, compute and output the value of f, with only **two digits** in the decimal part all the time.

Note: Your code should **NOT** use the function **pow(x,i)**.

Example 1 (Inputs are underlined):

```
Input x: 0.3
Input n: 5

Value is: 0.43
```

Example 2 (Inputs are underlined):

```
Input x: 0.1
Input n: 7

Value is: 0.11
```

Example 3 (Inputs are underlined):

```
Input x: 0
Input x: 1.2
Input x: 0.95
Input n: -1
Input n: 4
Input n: 5

Value is: 4.30
```

Question 4 [20 marks]

Write a program according to the requirements:

- Read 10 integers as the inputs and store them in an array first (**int arr[10]**). We assume that the input positive integer (data type: **int**) is valid. No need to check its correctness.
- We assume that the int type is large enough for the input, but we do not assume that your program knows the number of digits that each input number contains before the user's input.

- Calculate the sum of digits for each input number, and store the results in an output array (**int out[10]**). For example, for an input number 361, its sum of digits is 3+6+1=10, and 10 will be stored in the output array for input 361.

- Implement the Bubble sorting algorithm by yourself to sort all the input numbers in the array **arr[]** in an **increasing order**. During sorting, if two numbers in **arr[]** are swapped, their corresponding values in **out[]** should be swapped as well. In other words, after sorting, each **out[i]** still indicates the sum of digits for **arr[i]**.
- Display the sorted input numbers together with their corresponding results.

Example 1 (Inputs are underlined):

```
Enter all the numbers (arr):
12 23 34 23 56 34 89 10 361 204
Sum of digits for each input number (out):
3 5 7 5 11 7 17 1 10 6

Sorted arrays:
10 12 23 23 34 34 56 89 204 361
1 3 5 5 7 7 11 17 6 10
```

Example 2 (Inputs are underlined):

```
Enter all the numbers (arr):
32 47 10 2 83 59 94 195 204 7612
Sum of digits for each input number (out):
5 11 1 2 11 14 13 15 6 16

Sorted arrays:
2 10 32 47 59 83 94 195 204 7612
2 1 5 11 14 11 13 15 6 16
```

Question 5 [20 marks]

Assume there are 10 books and their IDs are represented by the characters from 'A' to 'J'. Please Implement a class called Book, which contains five **private** member variables:

- **id**: a single **char** variable, storing the book ID
- **subject**: a **string object**, storing the subject this book belongs to, e.g., "Computer Science"
- **price**: an **int** variable, storing the price of the book
- **month**: an **int** variable, storing the month for the **returning date** of the book
- **day**: an **int** variable, storing the day for the **returning date** of the book

The class Book also has the following **public member functions**:

- **getID()**: returns the book's ID
- **getSub()**: returns the book's subject
- **getPrice()**, returns the book's price
- **getMonth()**, returns the book's returning month
- **getDay()**, returns the book's returning day
- **set(char i, string s, int p, int m, int d)**, sets each member variable

Part-a): Complete the class definition of Book (do **NOT** change any identifiers given above and do **NOT** add any extra members).

- All the **member variables** are **private**, and all the **member functions** are **public**
- Declare all the **member variables**
- Implement getID(), getSub(), getPrice(), getMonth() and getDay() inside the class directly
- Write the prototype of set(...) inside the class. Implement it outside the class.

Part-b): In Q5(), we provide the information for ten books, including their IDs, subjects, prices and returning dates. Please declare an array to store 10 Book objects, and use the information provided above to initialize each Book object. Then print the information of each book as shown in the expected output.

Next, implement a **non-member** function, sortBooks(Book bArr[]). It sorts an array of Book objects, according to their prices in an **ascending** (**increasing**) order. If two books have the same price, further sort them according to their subjects in an **descending** (**decreasing**) order.

Note: you can use the compare() function defined in the <string> library to compare the **subjects** (**string objects**) of two books for sorting.

Expected Output:

Example:

```
The list of the books:
Book A: Math 2-20
Book B: Computer Science 3-16
Book C: Biology 2-23
Book D: Computer Science 10-5
Book E: Physics 11-2
Book F: Psychology 1-1
Book G: Business 5-30
Book H: Physics 6-24
Book I: Chemistry 8-17
Book J: Biology 5-8

The list of sorted books:
B(60, Computer Science)
A(80, Math)
D(80, Computer Science)
J(80, Biology)
C(150, Biology)
I(160, Chemistry)
E(200, Physics)
H(240, Physics)
G(240, Business)
F(300, Psychology)
```