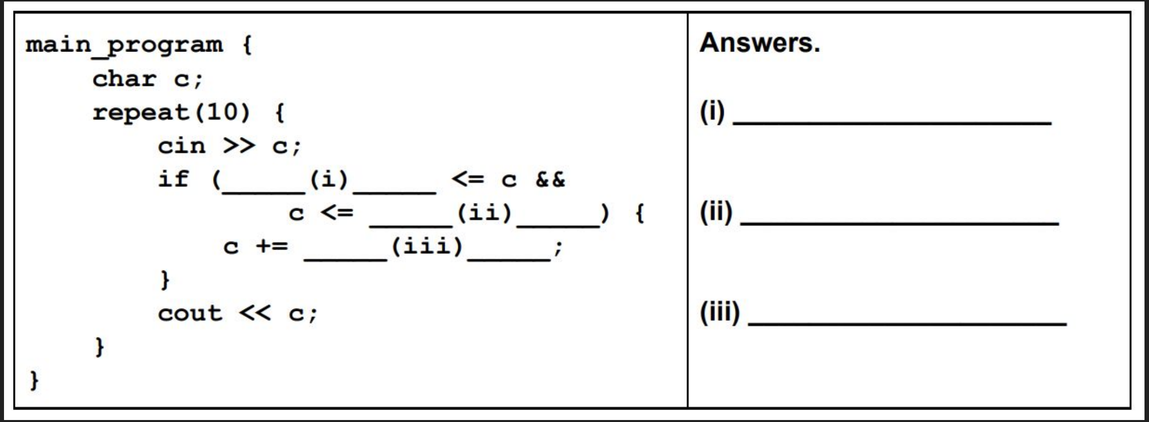
CS101 - Midsem Practice

Spring 2023-24

Swapnoneel Kayal & Maithri Suresh
Prof Shivaram Kalyanakrishnan

Question 1:

The following code reads 10 characters, one by one, and converts all lowercase alphabet to uppercase. It doesn't change other characters. Fill in the blanks to complete the code.
Note: In the ASCII mapping, lowercase characters are mapped sequentially i.e. one after the another. Similarly, for uppercase characters. Note that the difference between ASCII of lowercase 'a' and uppercase 'A' is not 26.



```
main_program {
    char c;
    repeat(10) {
        cin >> c;
        if (_____(i)_____ <= c &&
                c <= _____(ii)_____) {
            c += _____(iii)_____;
        }
        cout << c;
    }
}
```

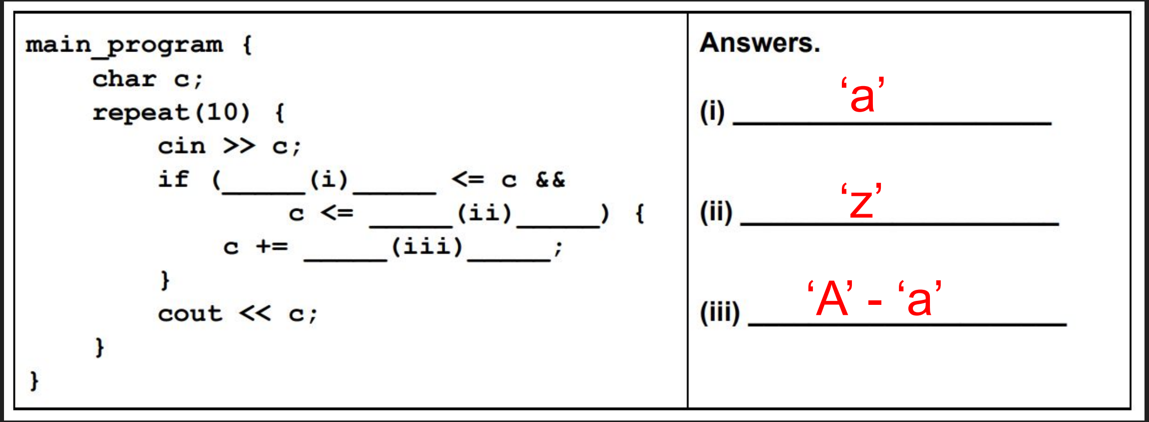
Answers.

(i) ____________________

(ii) ____________________

(iii) ____________________

Solution 1:



```
main_program {
    char c;
    repeat(10) {
        cin >> c;
        if (_____(i)_____ <= c &&
                c <= _____(ii)_____) {
            c += _____(iii)_____;
        }
        cout << c;
    }
}
```

Answers.

(i) ‘a’

(ii) ‘z’

(iii) ‘A’ - ‘a’

Alternate solution: (i) 97 (ii) 122 (iii) -32.

Similarly, some characters may be replaced by their ASCII codes, hence there are 8 possible combinations.

(1) 'a' <= c && c <= 'z': This condition checks if the character c is a lowercase letter by comparing it with the ASCII values of 'a' and 'z'. If c falls within this range, it indicates that c is a lowercase letter.

(2) c += 'A - 'a': If the condition 'a' <= c && c <= 'z' is true, it means c is a lowercase letter. To convert it to uppercase, we need to subtract the ASCII difference between lowercase 'a' and uppercase 'A' from c. This effectively converts the lowercase letter to its corresponding uppercase letter.

Question 2:

Given below are four programs. In each case predict the output.

Program - i

```
void incr(int &a) {
        a++;
}
main_program {
int a = 10;
incr(a);
cout << a;
}
```

Program - ii

```
void incr(int b) {
        int a = b;
a++; }
main_program {
int a = 10;
incr(a);
cout << a;
}
```

Program - iii

```
int incr(int a) {
        int b = a;
        b++;
        return a;
}
main_program { int
a = 10;
a = incr(a);
cout << a;
}
```

Program - iv

```
int  incr(int b) {
        int a = b;
        a ++;
        return a;
}
main_program {
int a = 10;
a = incr(a);
cout << a;
}
```

Solution 2:

(i) 11 [The function incr modifies the value of a by incrementing it by 1, and a is passed to the function by reference, so the modification affects the original variable a declared in main_program]
(ii) 10 [The incr function operates on a copy of a, not the original a, so the increment (a++) does not affect the original variable a in main_program]
(iii) 10 [The function incr takes an integer a, increments a local variable b by one, and returns the unchanged value of a, hence the output remains 10]
(iv) 11 [The function incr increases the value of its input by 1 before returning it, so when incr(a) is called where a = 10, it returns 11, thus outputting 11 when cout << a is executed in the main_program.]

Question 3:

What is the output of the following code segment?

```
for(unsigned int i = 1; i <= 10; i++){
    for(unsigned int j = -i; j < i; j++)
        cout << j << " ";
    cout << endl;
}
```

Solution 3:

What is the output of the following code segment?

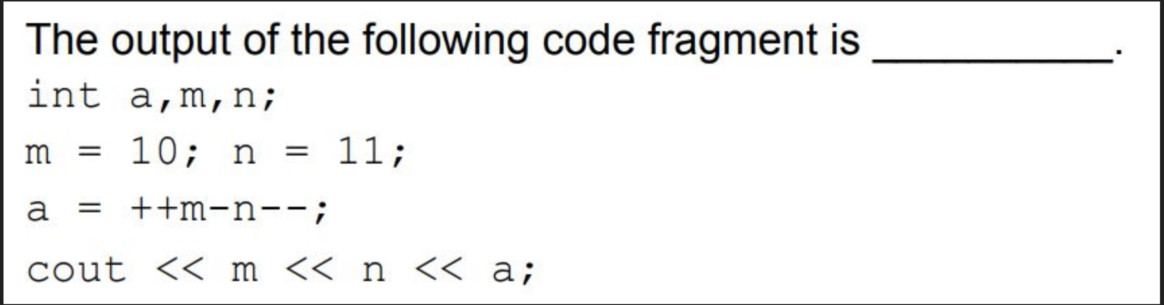
```
for(unsigned int i = 1; i <= 10; i++){
   for(unsigned int j = -i; j < i; j++)
      cout << j << " ";
   cout << endl;
} // <10 endl’s (blank lines)>
```

a very large number!

unsigned int usage

Question 4:

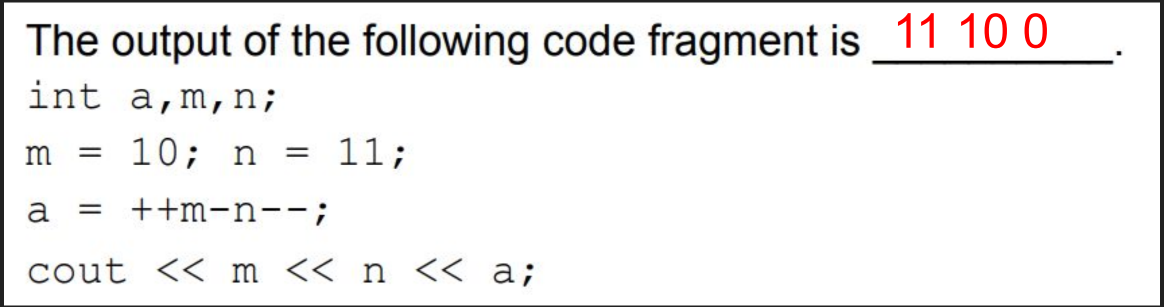
Given below is a program. Predict the output.


The output of the following code fragment is __________.

```
int a,m,n;
m = 10; n = 11;
a = ++m-n--;
cout << m << n << a;
```

Solution 4:

Given below is a program. Predict the output.



The output of the following code fragment is 11 10 0___________.

```
int a,m,n;
m = 10; n = 11;
a = ++m-n--;
cout << m << n << a;
```

m is pre-incremented so m ← m + 1 = 11
n is post-decremented so n ← n = 11
Therefore, a ← 11 - 11 = 0
After a is assigned, n ← n - 1 = 10

Question 5:

What is the output of the following code segment if the input is 10?

```
#include<simplecpp>
main_program{
    int sum=100,n;
    cin >> n;
    repeat(n){
        int sum = sum + 1;
    }
    cout << sum << endl;
}
```

Solution 5:

What is the output of the following code segment if the input is 10?

Scope of Variables

```
#include<simplecpp>
main_program{
    int sum=100,n;
    cin >> n;
    repeat(n){
        int sum = sum + 1;
    }
    cout << sum << endl; // 100
}
```

Defining a new "sum" variable - not the same as our original sum!

Question 6: Approach?

Given an integer n, you will be given n integers as input. Among the n integers, there is one number which occurs twice and all the numbers are guaranteed to belong to [1,n]. You have to output the number which occurs twice and the one which is absent. Note: Use of array is not allowed.

Input Format: The first line contains an integer n>=2, the number of inputs. The next n lines contain the input integers.
Output Format: Two integers separated by a space.
The first in refer should be the one which appears twice followed by the one which is absent.

Sample Input : 5 2 1 4 2 5
Sample Output : 2 3
Sample Input : 2 1 1
Sample Output : 1 2

Solution 6:

Approach 1:

Let ‘a’ be the extra number and ‘b’ be the missing number. Calculate the sum of the given numbers, we will call this S_1 and sum of squares of given numbers, we will call this S_2.

Now :

Sum of first ‘n’ natural numbers = (n)*(n+1)/2

Sum of squares of first ‘n’ natural numbers = (n)*(n+1)*(2n+1)/6

What do we have :

(1) S_1 - (n)*(n+1)/2 = a - b

(2) S_2 - (n)*(n+1)*(2n+1)/6 = $a^2 - b^2$ = (a - b)*(a + b)

Other approaches?

Question 7:

What is the output of the following code segment?

```
# include < iostream >
using namespace std;
int main ()
{
    int a = 5;
    int b = 9;
    int sum , product ;
    sum = a + b;
    product = a * b;
    if (sum = product ) {
        cout << "Sum of " << a << " and " << b;
        cout << " equals their product ." << endl ;
    }
    else {
        cout << "Sum of " << a << " and " << b;
        cout << " is not equal to their product ." << endl ;
    }
    return 0;
}
```

Solution 7:

What is the output of the following code segment?

```
# include < iostream >
using namespace std;
int main ()
{
    int a = 5;
    int b = 9;
    int sum , product ;
    sum = a + b;
    product = a * b;
    if (sum = product ) {
        cout << "Sum of " << a << " and " << b;
        cout << " equals their product ." << endl ;
    }
    else {
        cout << "Sum of " << a << " and " << b;
        cout << " is not equal to their product ." << endl ;
    }
    return 0;
}
```

Notice that here we have = and not ==. This is NOT a syntax error, and will compile perfectly! This assigns the value of product to sum, and is treated as `true if it is non-zero.`

```
# include < iostream >
using namespace std;
int main ()
{
    int a = 5;
    int b = 9;
    int sum , product ;
    sum = a + b;
    product = a * b;
    if (sum = product ) {
        cout << "Sum of " << a << " and " << b;
        cout << " equals their product ." << endl ;
    }
    else {
        cout << "Sum of " << a << " and " << b;
        cout << " is not equal to their product ." << endl ;
    }
    return 0;
}
```

Solution 7:

What is the output of the following code segment?

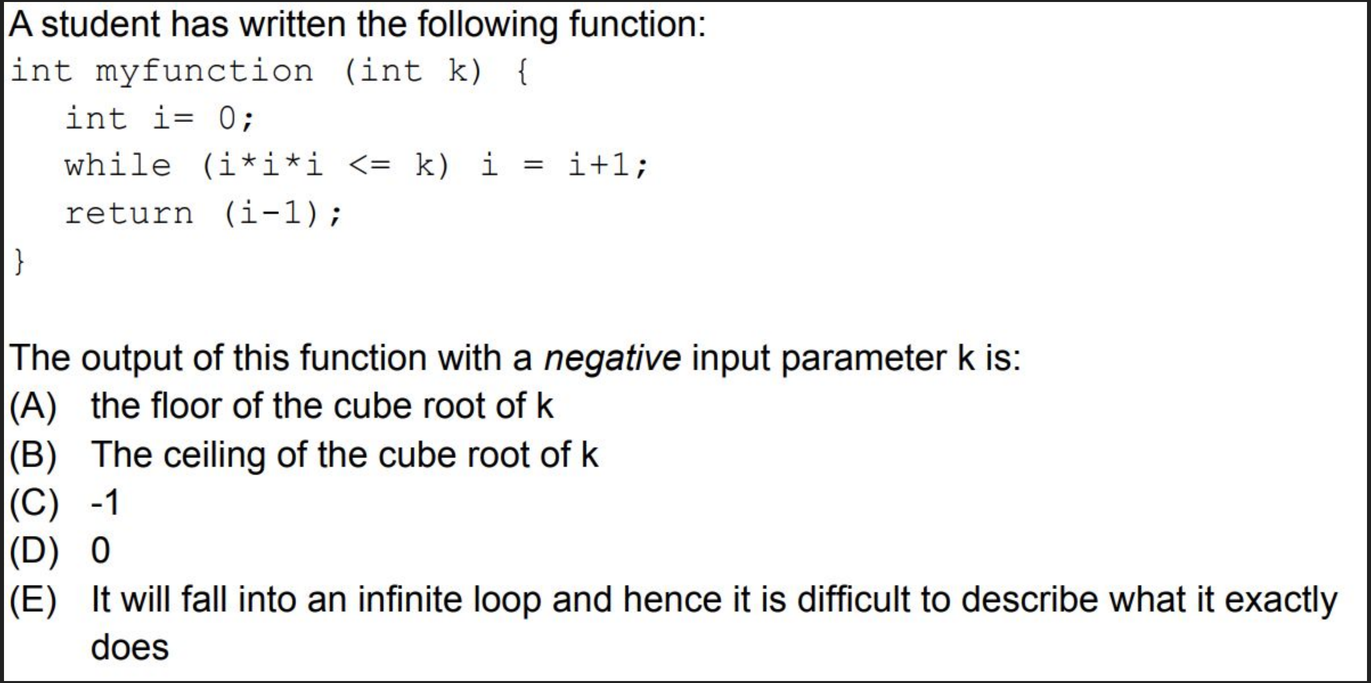
Output:

```
Sum of 5 and 9 equals
their product.
```

If-else conditions

L-value and R-value

Question 8:



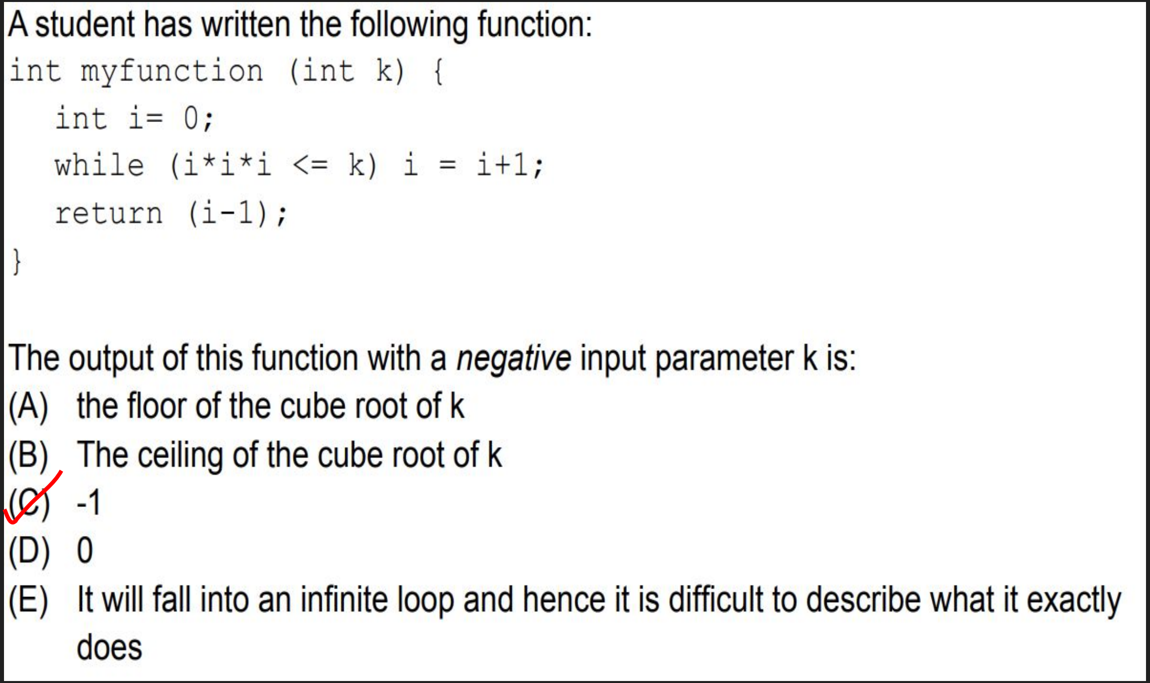
A student has written the following function:

```
int myfunction (int k) {
   int i= 0;
   while (i*i*i <= k) i = i+1;
   return (i-1);
}
```

The output of this function with a *negative* input parameter k is:

(A) the floor of the cube root of k
(B) The ceiling of the cube root of k
(C) -1
(D) 0
(E) It will fall into an infinite loop and hence it is difficult to describe what it exactly does

Solution 8:



A student has written the following function:

```
int myfunction (int k) {
  int i= 0;
  while (i*i*i <= k) i = i+1;
  return (i-1);
}
```

The output of this function with a *negative* input parameter k is:

(A) the floor of the cube root of k

(B) The ceiling of the cube root of k

[✓ (C) -1]

(D) 0

(E) It will fall into an infinite loop and hence it is difficult to describe what it exactly does

For a negative k, the flow of program is not able to execute (i ← i + 1) hence ‘i’ is stuck at 0. Hence, we get back (0-1 = -1) as the output.

Question 9:

What are the values of a, b and c at the end of the execution of the following code?

```
int a=11, b=22, c;
c = a + b + a++ + b++ + ++a + ++b;
```

Question 9:

What are the values of a, b and c at the end of the execution of the following code?

```
int a=11, b=22, c;
c = a + b + a++ + b++ + ++a + ++b;
//11 + 22 + 11 + 22 + 13 + 24

a=13, b=24, c=103
```

Post and Pre increment

All the best!