1. (12%) Suppose your program has the following declarations to represent information about a book:

```
string title;
int year_published;
bool paperback; //true if paperback, false if hardcover
```

Write C++ logical conditions corresponding to each of the following sets. Your answers should be as compact as possible and cover all cases.

(a) All paperbacks published after 1995 and before 2006.

```
paperback && year_published > 1995 && year_published < 2006
```

(b) All hardcover books, whose title starts with the letter 'A' or letter 'T', published in 2008 or later.

```
!paperback && year_published >= 2008 && (title[0] == 'A'|| title[0] == 'T')
```

(c) All hardcover books, published last year, whose title ends with the letter 's' ("Birds", "Luminaries", etc).

```
!paperback && title[title.length() - 1] == 's' && year_published == 2008
```

2. (10%) Write a C++ function that calculates: $\dfrac{(1 + n)^k}{\sqrt{k + 1}}$

```
#include <cmath>
double foo(double n, double k)
{
    return pow(1 + n, k) / sqrt(k + 1));
}
```

3. (18%) Consider the following program fragment:

```cpp
void foo(int v, int & r);

int main()
{
    int v = 0;  //SPECIAL LINE
    int r = 0;
    for (int i = 0; i < 2; i++)
    {
        foo(v, r);
        cout << v << " " << r << endl;
    }
    return 0 ;
}

void foo(int v, int & r)
{
    static int s = 0;
    s = s + 1;
    v = v + s;
    r = r + s;
}
```

(a) What does the program output?

```
0 1
0 3
```

(b) Underline all formal parameters in the program.

(c) Circle all actual arguments in the program.

(d) Draw a dashed box around all prototypes in the program.

(e) Draw a solid box around the scope of the variable declared on //SPECIAL LINE ?

(f) What is the value of variable s at the end of program execution - just before the main() function returns?

2

4. (15%) Write a function: **void remove_e(string & sentence)** that removes all occurrences of letter **'e'** from string **sentence** in place: in its original memory location in the caller function.

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

void remove_e(string & sentence);

int main()
{
    string sentence = "Hello hello";
    remove_e(sentence);
    cout << endl << sentence << endl;
    return 0;
}

void remove_e(string & sentence) {
    for (int i = 0; i < sentence.length(); i++)
    {
        if (sentence[i] == 'e') {
            sentence = sentence.substr(0, i) + sentence.substr(i + 1,
sentence.length() - 1);
        }
    }
}
```

5. (15%) Write a program that asks user for a positive integer side length. If they enter an illegal value, they must be prompted to enter a good one until they do. It then displays a rectangle of the given side length, made up of alternating horizontal lines of o's and x's. For example, if the side length is 5, the program should display:

```
O O O O O
X X X X X
O O O O O
X X X X X
O O O O O
```

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

int main()
{
    int side;

    do {
        cout << "Enter side length: ";
        cin >> side;
    } while (side < 0);

    for (int i = 0; i < side; i++)
    {
        for (int j = 0; j < side; j++)
        {
            if (i % 2 == 0)
                cout << "O";
            else
                cout << "X";
        }
        cout << endl;
    }
    return 0;
}
```