Problem (1) [10 points] Make a function that takes as an input a character array and an array of integers. This function should return a dynamically created string array, created by combining the char array into words of length denoted by the int array.

More specifically, the first element in the integer array denotes how many characters from the character array comprise the first word. The second element in the integer array indicates that those many letters after the first word in the character array are the second word. And so on....

Example:
```
char words[] = {'h', 'i', 't', 'h', 'e', 'r', 'c', 'l', 'a', 's', 's'};
int lens[] = {2, 5, 5};
// The above input into your function should return a string array:
// {"hi", "there", "class"}
```

```
string* chopchop(char letters[], int lens[], int arrLen)
{
    string* result = new string[arrLen];
    int spot=0;
    for(int i=0; i < arrLen; i++)
    {
        for(int j=0; j < lens[i]; j++)
        {
            result[i] += letters[spot];
```
spot++;
}
}

return result;
}
Problem (2) [10 points] Write the output of the following code:

class A {
    private:
        int num;
        float fl;
    public:
        A( );
        int getInt( );
        float getFloat( );
        ~A( );
    };

    int A::getInt( ) {
        return num;
    }

    float A::getFloat( ) {
        return fl;
    }

    A::A( ) {
        cout << "Initializing default" << endl;
        num=5;
        fl=2.5;
    }

    A::~A( ) {
        cout << "Destructor is active" << endl;
    }

    void main( ) {
        A array[2];
        cout << array[1].getInt() << " " << array[1].getFloat() << endl;
    }

    Initializing default
    Initializing default
5 2.5
Destructor is active
Destructor is active
Problem (3) [10 points] Find 5 errors (not three) in this code. Describe clearly why there is an error and how you would fix it. You may assume all #includes are done correctly and it is “using namespace std”. All other code is shown.

class A {
private:
    int m_iA;
    string m_sA;
public:
    A();
    void PrintIntegerA();
};

A::A() {
    m_iA = 0;
    m_sA = ABC;
}

class B : public A {
private:
    int m_iB;
public:
    void PrintIntegerB();
};

void B::PrintIntegerB() {
    m_iB = m_iA;
    cout << m_iB << endl;
}

int main() {
    A* a;
    A->PrintIntegerA();

    B b;
    b->PrintIntegerB();
}

1. " m_sA = ABC;", "ABC" needs quotes to make it a string value, otherwise it is interpreted as a variable
2. " m_iB = m_iA;" "m_iA" is private so B cannot access,
instead "m_iA" should be protected
3. "A->PrintIntegerA();" "A" is the class, not a specific variable (instance). Needs to be "a->PrintIntegerA();"
4. "a->PrintIntegerA();" "a" does not point to anything, this will crash. Need to say "a = new A();" after the declaration.
5. The "PrintIntegerA()" function is never defined... this needs to be done like: "void A::PrintIntegerA() { ... }".
6. "b->PrintIntegerB();" "b" is not a pointer, so it should not use "->". Should be "b.PrintIntegerB();"
**Problem (4) [10 points]** Using the following Person class, create two child/subclasses called Student and Professor. In each of the child/subclasses, create useful functions for getData() and outstanding(). The Student class should contain a CGPA (grade points in double) and the Professor subclass should contain number of publications (int). A student is outstanding if his/her cgpa is more than 3.7 and a professor is outstanding is her/his publications are more than 50. Make the getData() function cin any additional info needed to implement the functions above correctly.

```cpp
class Person {
    protected:
        string name;
    public:
        void getName();
        virtual void getData(){}
        virtual bool outstanding(){}
};

void Person::getName() {
    cin >> name;
}

class Student : public Person {
    protected:
        double CGPA;
    public:
        void getData();
        bool outstanding();
};

void Student::getData() {
    cin >> CGPA;
}

bool Student::outstanding() {
    return CGPA > 3.7;
}

class Professor : public Person {

protected:
    int numPubs;
public:
    void getData();
    bool outstanding();
};

void Professor::getData()
{
    cin >> numPubs;
}

bool Professor::outstanding()
{
    return numPubs > 50;
}
**Problem (5)** [10 points] Create a tickTock class to make the following main() function work for the example shown:

```cpp
int main() {
    tickTock tt;
    cout << "Enter the hours, minutes and seconds:" << endl;
    cin >> tt;
    cout << tt;
}
```

Example (user input is underlined):
Enter the hours, minutes and seconds:

4 55 21
4:55:21

```cpp
class tickTock
{
private:
    int hour;
    int min;
    int sec;
public:
    friend istream& operator>>(istream& in, tickTock &tt);
    friend ostream& operator<<(ostream& out, tickTock tt);
};
```

```cpp
istream& operator>>(istream& in, tickTock &tt)
{
    in >> tt.hour >> tt.min >> tt.sec;
    return in;
}
```

```cpp
ostream& operator<<(ostream& out, tickTock tt)
{
    out << tt.hour << ":" << tt.min << ":" << tt.sec;
    return out;
}
```
Problem (6) [10 points] The class below is only partially complete. Complete the class (you only need to write function definitions) to do two things:
(1) Make the class work with main(), where the “cpp” variable should be a class of 240 students called “csci-1113-001”,
(2) Not leak any memory.

```cpp
struct UoMStudents {
    string name;
    string grade;
};

class UoMClass {
private:
    string className;
    int classSize;
    UoMStudents *students; // dynamic array of students
public:
    // blank right now...
};

int main() {
    UoMClass cpp(240, "csci-1113-001");
}
```

```cpp
UoMClass::UoMClass(int numStu, string name) {
    className = name;
    classSize = numStu;
    students = new UoMStudents[classSize];
}

UoMClass::~UoMClass()
{
    delete [] students;
}
```
Problem (7) [10 points] What is the output of the following code:

```cpp
int x = 1;
int y = 2;
int* a = &x;
int* b = a;
int** c = &b;
(*b)++;
b = &y;
**c = 2 + *b;
a = *c;
*a = *b + **c;

cout << x << " " << y << " " << *a << " " << *b << " " << **c << endl;
```

2 8 8 8 8
Problem (8) [10 points] Explain specifically the problem with each of these code fragments:

(1)
string* theBest = new string("Mina");
cout << *theBest << " skips grading."
theBest = new string("Alex");

(2)
int* xptr;
*xptr = 2;

(3)
int* randArray(int n) {
    int* arr = new int[n];
    for(int i=0; i < n; i++)
    {
        arr[i] = rand();
    }
    delete[] arr;
    return arr;
}

(4)
char* alph[26];
for(int i=0; i < 26; i++)
{
    alph[i] = new char;
}
// do some stuff
delete[] alph;

(1) Memory leak.
Have to say: "delete theBest" before the line:
"theBest = new string("Alex");"
(2) xptr never points anywhere (uninitialized),
so it will (probably) crash when you try to do "*xptr = 2;"
(3) Array is deleted yet returned.
This means whoever wishes to use this returned array cannot (as it is deleted).
(4) "alph" is an array of pointers.
You must delete each pointer individually,
so the delete should also be in a loop just like the new (and "delete alph[i]").
Problem (9) [10 points] Write (in C++) a function that takes four integers as input corresponding to two \((x, y)\) points. Return whether or not these two points are perpendicular to each other. Two points are perpendicular if: \(x_1 \cdot x_2 + y_1 \cdot y_2 = 0\).

```cpp
bool perp(int x1, int y1, int x2, int y2)
{
    return x1*x2 + y1*y2 == 0;
}
```
Problem (10) [10 points] Open a file called “helloWorld.txt” and display the 100th word in the file. If there is not 100 words in the file, say “File is too small”.

    ifstream reader;
    reader.open("helloWorld.txt");
    int lineNum = 0;
    string line;
    getline(reader, line);
    while(!reader.eof())
    {
        lineNum++;
        if(lineNum == 100)
        {
            break;
        }
        getline(reader, line);
    }
Problem (11) [10 points] What is the output of the following program:

```cpp
bool foo(int x);
bool bar(int x);

int main() {
    if( (foo(4) || bar(1)) && (bar(-2) && foo(3)) )
    {
        foo(9);
    } else
    {
        bar(9);
    }
}

bool foo(int x) {
    cout << x; // no endl
    return x > 3;
}

bool bar(int x) {
    cout << -x; // no endl
    return true;
}
```

423-9
**Problem (12)** [10 points] Make (in C++) a replaceChar() function that takes as input a string and two characters and replaces everywhere where the first character appears in the string with the second character.

Example:

```cpp
string word = "cookie";
replaceChar(word, 'o', 'y');
cout << word; // shows: cyykie
```

```cpp
void replaceChar(string& w, char ori, char repla)
{
    for(int i=0; i < w.length(); i++)
    {
        if(w[i] == ori)
        {
            w[i] = repla;
        }
    }
}
```