Question 1: [20 points]

Write a C++ program which will prompt the user for an integer amount of money in cents and will calculate the MINIMUM amount of change needed to make up this amount. Assume that the following coins exist:

two dollar coin (twonnie)
one dollar coin (loonie)
quarter
nickel
dime
penny

A sample interaction with this program would be as follows:

Enter the amount as an integer number of cents 345
Twonnie = 1
loonie = 1
quarter = 1
dime = 2
nickel = 0
penny = 0

Question 2: [25 points]

A circular shift of an array moves the elements of an array one position to the left or to the right with either the first or last element of the array wrapping around. For example, if we have a 5 element integer array whose initial contents are:

1 8 2 4 3

if we shift right by one position the array becomes:

3 1 8 2 4

if we shift left the original array it becomes:

8 2 4 3 1

The shift left and right operations can be extended into rotate operations which perform multiple shifts. For example, we can rotate the original array to the right by 0,1,2,3,4 or 5 places to obtain:

1 8 2 4 3 (original array)
3 1 8 2 4 (rotate right by 1 place)
4 3 1 8 2 (rotate right by 2 places)

2 4 3 1 8 (rotate right by 3 places)
8 2 4 3 1 (rotate right by 4 places)
1 8 2 4 3 (rotate right by 5 places)

The corresponding rotate left operations would behave similarly except that the array contents would move in the opposite direction:

1 8 2 4 3 (original array)
8 2 4 3 1 (rotate left by 1 place)
2 4 3 1 8 (rotate left by 2 places)
4 3 1 8 2 (rotate left by 3 places)
3 1 8 2 4 (rotate left by 4 places)
1 8 2 4 3 (rotate left by 5 places)

Write C++ functions called **shift_right** and **shift_left** which receive an integer array and the size of the array as parameters. The functions perform a circular shift right and left of the array by one position.

Next, write C++ functions called **rotate_right** and **rotate_left** which receive an integer array, the size of the array, and an integer called count which determines the number of places to rotate the array by. These functions perform the appropriate rotate of the passed array.

Write a main() function which defines an initializes an array of 5 integer and then invokes the rotate_left and rotate_right functions 5 times each using a value of 0,1,2,3,4,5 for the count argument.

Question 3: [15 points]

(a) What would the output of the following program be:

```
#include <iostream>
#include <string>

using namespace std;

int counter = 0;

void f()
{
 ++counter;
 cout << “inside f()” << endl;
}

void g()
{
```

```
 f();
 f();
 cout << “inside g()” << endl;
}

void h()
{
 f();
 g();
 f();
 cout << “inside h()” << endl;
}


int main()
{

f();
cout << counter << endl;
g();
cout << counter << endl;
h();
cout << counter << endl;

return 0;
}
```

(b) What would the output of the following program be:

```
#include <iostream>
#include <string>

using namespace std;

int counter = 0;

void f()
{
 ++counter;
 cout << “inside f()” << endl;
}

void g()
{
 f();
```

```
 counter--;
 f();
 cout << “inside g()” << endl;
}

void h()
{
 f();
 g();
 f();
 counter--;
 cout << “inside h()” << endl;
}


int main()
{

f();
cout << counter << endl;
g();
cout << counter << endl;
h();
cout << counter << endl;

return 0;
}
```

Question 4: [15 points]

What would the output of the following program be:

```
#include <iostream>
#include <string>

using namespace std;


int count = 0;

int main()
{

int count = 5;
count++;
cout << count << endl;
```

```
{
 int count = 2;
 if (count > 3)
   cout << count << endl;
 else
  {
    int count = 10;
   cout << “not bigger than 3 “ << endl;
   cout << count << endl;
  }
}
::count++;
cout << ::count << endl;

return 0;
}
```

Question 5: 25 points]

The words ‘eve’, ‘elle’, ‘toot’, ‘rotor’ are called PALINDROMES since reversing the order of the letters contained in them does not change them. The words ‘ted’, ‘mick’, ‘keith’ are NOT palindromes since ‘ted’ does not equal ‘det’, nor does ‘mick’ equal ‘kcim’.

Write a non-recursive function which returns a bool set to true if the array which is passed to the function represents a palindrome and false otherwise. Pass an integer argument as a second parameter which specifies the size of the passed character array.

Write a main program which declares and initializes 5 character arrays to contain the characters:

‘e’ ‘v’ ‘e’
‘e’ ‘l’ ‘l’ ‘e’
‘t’ ‘o’ ‘o’ ‘t’
‘r’ ‘o’ ‘t’ ‘o’ ‘r’
‘m’ ‘a’ ‘d’ ‘a’ ‘m’ ‘i’ ‘m’ ‘a’ ‘d’ ‘a’ ‘m’

Your function should NOT copy the passed array to a temporary array in order to verify whether it is a palindrome or not. The test for the arrays ‘palindromeness’ should be performed in place (directly on the original passed array).