PROGRAMMING FUNDAMENTALS

ASSIGNMENT NO: 02

SUBMITTED TO: MISS AASMA AKRAM

SUBMITTED BY: LINTA AFFAF

DEPARTMENT: BSSE

SEMESTER: 01(self)

4.11)

a) In this exercise, we need to find errors in the given code. The first error is in *line 1*, a semicolon is not written after **if statement**. The second error is in *line 6*, if we want a new line, we need to write *endl* outside of the string.

Correct code is:

```
if (age >= 65) {
    cout << "Age is greater than or equal to 65" <<endl;
}else {
    cout << "Age is less than 65 << endl";
return o;
}
```

b) In this exercise, we need to find errors in the given code. The first error is in *line 4*, a semicolon is not written after *else statement*. The second error is in line 5, if we want a new line, we need to write endl outside of the string.

Correct code is:

```
if (age >= 65) {
    cout << "Age is greater than or equal to 65" <<endl;
}else {
    cout << "Age is less than 65" << endl;
return 0;
}
```

c) In this exercise, we need to find errors in the given code. In this code, the variable **total** is not initialized with a value. It is a good practice to assign value to variables.

```
unsigned int x{1};
unsigned int total{0};

while (x <= 10) {
    total += x;
    ++x;
Return 0;
}
```

d) In this exercise, we need to find errors in the given code. When we use **while statement** and have multiple statements inside the **while statement**, we need to use parenthesis. Also, **while** doesn't start with a capital letter.

```
while(x <= 100){
    total = total + x;
    ++x;
Return 0;
}
```

e) In this exercise, we need to find the errors in the given code. In this code, there is one logical error. If **y** is greater than 0, the code will never end because it will stuck in *while* loop forever. If **y** is less than 0, it will never enter the loop.

4.12)

In this exercise, the given code takes **x**, multiplies it by itself, stores it into **y**, and outputs the value of **y**. After that, **y** is added to **total**, and **x** is increased by 1. These steps repeat until **x** is greater than 10. At the end, **total** is printed.

Output of code:

```
1
4
9
16
25
36
49
64
81
100
Total is 385
```

4.13)

Step 1

In this exercise, we need to write pseudocode and a program that prompts the user to enter mileage and gallons per trip. Prompt user to enter these values until -1 is entered for mileage. Then, calculate used gallons per mile and total gallons per mile and output the results.

Step 2

The pseudocode

```
Determine miles per gallon per trip and in total
Initalize the variables
Prompt user to enter first mileage
Input the first mileage (possibly the sentinel)
While the user has not entered the sentinel
    Prompt user to enter used gallons
    Add miles to total miles
    Add gallons to total gallons
    Calculate and output miles per gallon for this trip
    Calculate and output combined miles per gallon up to
    this point
    Prompt user to enter mileage
    Input next mileage (possibly the sentinel)
```

Step 3

#include

```
using namespace std;

int main(){
    double miles = 0;
    double gallons = 0;
    double totalMiles = 0;
    double totalGallons = 0;

    cout << "Enter miles driven (-1 to quit): " << endl;
    cin >> miles;

    while(miles != -1){
        cout << "Enter gallons used: " << endl;
        cin >> gallons;
        totalMiles += miles;
        totalGallons += gallons;
        cout << "MPG this trip: " << miles/gallons
             << endl;
        cout << "Total MPG: "
             << totalMiles / totalGallons << endl;
        cout << "\n\nEnter miles driven (-1 to quit): "
                << endl;
        cin >> miles;
    }
Return 0;
}
```

4.15)

Step 1

In this exercise, we need to write a program that calculates the salesperson's salary depending on its total sales this month. The program asks the user to enter total sales, calculates and outputs the salary.

Step 2

Let's first write a pseudocode.

```
Determine salesperson's salary
Initalize the variables
Prompt user to enter sales in dollars
Input the first sales (possibly the sentinel)
While the user has not entered the sentinel
    Calculate salary as 200 + 0.09 * sales
    Output salary
    Prompt user to enter sales in dollars
    Input sales (possibly the sentinel)
```

Step 3

```
#include<iostream>

using namespace std;

int main(){
    double sales;

    cout << "Enter sales in dollars (-1 to end): ";
    cin >> sales;
    while(sales != -1){
        cout << "Salary is: $" << 200 + 0.09 * sales
             << endl;
        cout <<"\nEnter sales in dollars (-1 to end): ";
        cin >> sales;
return 0;
    }
}
```

4.16)

Step 1

In this exercise, we need to write a program that calculates salaries from given hours and hourly rates. The program prompts the user to enter hours and hourly rates for each employee. If the worker has worked more than 40 hours, the program must compensate extra hours with a bigger salary. First, we will multiply total hours with hourly rate and for every extra hour, add half of the rate per hour.

Step 2

pseudocode.

```
Determine salary
Initalize the variables
Prompt user to enter worked hours
Input the first hours (possibly the sentinel)
While the user has not entered the sentinel
    Prompt use to enter hourly rate
```

```
    Input hourly rate
    salary = hours * rate
    If hours > 40
        salary += 0.5 * (hours - 40)
    Output salary
    Prompt user to enter worked hours
    Input the hours (possibly the sentinel)
```

Step 3

```
#include<iostream>

using namespace std;

int main(){
    int hours;
    double rate;
    double salary;

    cout << "Enter hours worked (-1 to end): ";
    cin >> hours;
    while(hours != -1){
        cout << "Enter hourly rate: ";
        cin >> rate;
        salary = hours * rate;
        if(hours > 40){
            //Add extra for extra hours
            salary += 0.5 * (hours - 40) * rate;
        }
        cout << "Salary is $" << salary << endl;
        cout << "\nEnter hours worked (-1 to end): ";
        cin >> hours;
    }
}
```

4.17)

Step 1

In this exercise, we need to find the largest number among the entered numbers. The program prompts the user to enter 10 numbers and finds the largest number. To do that, the program uses **while** loop until the user has entered 10 numbers. For each number, we compare that number with the current largest number. If the entered number is greater than the current largest number, the entered number becomes the largest number. After the user has entered 10 numbers, the program outputs the largest number.

Step 2

```
#include<iostream>

using namespace std;

int main(){
    int counter = 0;
    int number;
    int largest = 0;
```

```
    while(counter < 10){
        cout << "Enter the number: " << endl;
        cin >> number;
        if(number > largest)
            largest = number;
        counter += 1;
    }
    cout << "Largest entered number is " << largest
         << endl;
}
```

4.18)

In this exercise, we need to write out the given table. The table is outputted with **while** statement. Following program outputs the given table.

```
#include<iostream>

using namespace std;

int main(){
    int N = 1;

    cout << "N\t10*N\t100*N\t1000*N\n";
    while(N <= 5){
        cout << N << "\t" << 10*N << "\t" << 100*N
             << "\t" << 1000*N <<"\n";
        N += 1;
    }
Return 0;
}
```

4.19)

Step 1

In this exercise, we need to write a program that finds the two largest numbers from given numbers. The program prompts the user to enter 10 numbers, and each number is compared with the largest number and with the second-largest number. If the entered number is greater than the largest number, the new largest number is the entered number. If the entered number is greater than the second-largest number but less than the largest number, the entered number becomes the second-largest number.

```
#include<iostream>

using namespace std;

int main(){
    int first = 0;
    int second = 0;
    int counter = 0;
    int number;
```

```
    while(counter < 10){
        cout << "Enter the number: " << endl;
        cin >> number;
        if(number > second){
            if(number > first){
                first = number;
            }
            if(number < first){
                second = number;
            }
        }
        counter += 1;
    }
    cout << "Largest number is " << first << endl;
    cout << "Second largest number is " << second
         << endl;
}
```

Question no 4.23

```
#include <iostream>

using namespace std;

int main() {

    int size = 5;

    for (int i = 0; i < size; ++i) {

    for (int j = 0; j < i; ++j) {

         cout << " "; }

     for (int k = 0; k < 2 * (size - i) - 1; ++k) {

            if (i == 0 || k == 0 || k == 2 * (size - i) - 2) {

              cout << "#";

     } else {

               cout << "*";

            }

        }

      cout << endl;

    }


    for (int i = size - 2; i >= 0; --i) {

        for (int j = 0; j < i; ++j) {

         cout << " ";
```

```
        }
        for (int k = 0; k < 2 * (size - i) - 1; ++k) {
            if (i == 0 || k == 0 || k == 2 * (size - i) - 2) {
              cout << "#";
            } else {
             cout << "*";
            }
        }
    cout << endl;
    }
     return 0;
    }
```

Question no 4.24

```
#include <iostream>
using namespace std;
int main() {
    int size;
 do {
      cout << "Enter an odd number in the range 1 to 29: ";
      cin >> size;
    } while (size < 1 || size > 29 || size % 2 == 0);

    for (int i = 0; i < size / 2 + 1; ++i) {
        for (int j = 0; j < i; ++j) {
        cout << " ";
        }
        for (int k = 0; k < 2 * (size / 2 + 1 - i) - 1; ++k) {
            if (i == 0 || k == 0 || k == 2 * (size / 2 + 1 - i) - 2) {
            cout << "#";
            } else {
      cout << "*";
```

```
            }
        }
    cout << endl;
    }
for (int i = size / 2; i >= 0; --i) {
        for (int j = 0; j < i; ++j) {
          cout << " ";
        }
        for (int k = 0; k < 2 * (size / 2 + 1 - i) - 1; ++k) {
            if (i == 0 || k == 0 || k == 2 * (size / 2 + 1 - i) - 2) {
              cout << "#";
            } else {
             cout << "*";
            }
        }
  cout << endl;
    }
 return 0;
}
```

Question no 4.25

```
#include <iostream>
Using namespace std;

int main() {
    bool exitLoop = false;

    for (int i = 0; i < 10 && !exitLoop; ++i) {
        if (/* some condition for early exit */) {
            exitLoop = true;
        }
 return 0;
}
```

Q.4.29:

(Checkerboard Pattern of Asterisks) Write a program that displays the following checkerboard pattern. Your program must use only three output statements, one of each of the following forms:

cout << "* ";

cout << ' ';

cout << endl;

```
********
 ********
********
 ********
********
 ********
********
 ********
```

```
#include<iostream>
using namespace std;
main()
{
	for(int row = 0; row < 8; ++row)
	{
	for(int col = 0; col < 8; ++col)
	{
	if((row + col) % 2 == 0)
	{
	cout<<"*";
	}
	else
	{
	cout<<" "; }
```

```
	}
	cout<<endl;
	}
	return 0;
	}
```

Q.4.30:

(Fibonacci Sequence) Write a program that prints the Fibonacci sequence 0, 1, 1, 2, 3, 5, 8, etc. Use 0 and 1 as your seed values. Each subsequent number in the Fibonacci sequence is the sum of the previous two numbers. Your while loop should not terminate(i.e., you should create an infinite loop). To do this, simply use the keyword **true** as the expression for the while statement. What happens when you run this program?

```
#include<iostream>
using namespace std;
main()
{
	int first = 0, second = 1, next;
	while (true)
	{
	cout<<first<<" ";
	next = first + second;
	first = second;
	second = next;
	}
	return 0;
	}
```

Q.4.31:

(Calculating a Sphere's Circumference, Area and Volume) Write a program that reads the radius of a sphere (as a double value) and computes the circumference, area and volume of the spheres. Use the value 3.14159 for π.

```
#include<iostream>
using namespace std;
main()
{
      const double pi = 3.14159;
      double redius;
      cout<<"Enter the radius of the sphere";
      cin>>radius;
      double circumference = 2*pi*radius;
      double area = 4*pi*(radius, 2);
      double volume = (4.0 / 3.0)*pi*(radius, 3);
      cout<<"Circumference="<<circumference<<endl;
      cout<<"Surface Area="<<area<<endl;
      cout<<"Volume="<<volume<<endl;
      return 0;
  }
```

Question 4.35

a)

```
#include <iostream>
using namespace std;
int main()
```

```
 int n;
cout << "Enter a nonnegative integer: ";
  cin >> n;
 int factorial = 1;
   while (n > 1) {
      factorial *= n;
      n--;
   }
   cout << "Factorial is: " << factorial << endl;
 return 0;
}
```

b)

```
#include <iostream>
using namespace std;
#include <cmath>
int main() {
   int accuracy;
cout << "Enter the desired accuracy for e estimation (number of terms): ";
cin >> accuracy;
double e = 1.0;
double term = 1.0;
 int i = 1;
   while (i <= accuracy) {
      term /= i;
      e += term;
      i++;
   }
cout << "Estimated value of e: " << e << endl;
 return 0;      }
```

C)

```
#include <iostream>
using namespace std;
#include <cmath>
int main() {
   double x;
   int accuracy;
 cout << "Enter the value of x: ";
cin >> x;
cout << "Enter the desired accuracy for e^x computation (number of terms): ";
cin >> accuracy;
  double result = 1.0;
   double term = 1.0;
   int i = 1;
   while (i <= accuracy) {
      term *= (x / i);
      result += term;
      i++;
   }
cout << "Result of e^x: " << result << endl;
return 0;
}
```

Question 3.56

```
#include <iostream>
using namespace std;
#include <iomanip>
class Account {
private:
```

```
string accountName;
    double balance;
public:
    Account(const  string& name, double initialBalance = 0.0) : accountName(name),
balance(max(initialBalance, 0.0)) {}

    void setAccountName(const string& name) { accountName = name; }
string getAccountName() const { return accountName; }

    void setBalance(double newBalance) { balance = max(newBalance, 0.0); }
    double getBalance() const { return balance; }

    void deposit(double amount) {
        if (amount > 0.0) cout << "Deposit of $" << fixed <<setprecision(2) << amount << " successful. New
balance: $" << (balance += amount) << endl;
        else cout << "Deposit amount must be greater than 0." << endl;
    }

    void withdraw(double amount) {
        if (amount > 0.0 && amount <= balance)  cout << "Withdrawal of $" << fixed << setprecision(2) <<
amount << " successful. New balance: $" << (balance -= amount) << endl;
        else  cout << "Invalid withdrawal amount or insufficient funds." << endl;
    }
};

int main() {
    Account myAccount("John Doe", 1000.0);
cout << "Account name: " << myAccount.getAccountName() << "\nInitial balance: $" << fixed <<
setprecision(2) << myAccount.getBalance() << endl;
```

```
  myAccount.deposit(500.0);
  myAccount.withdraw(200.0);
   return 0;
  }
```

Question 4.37

```
#include <iostream>
using namespace std;
int transform(int num, int modifier) {
    int digit;
    for (int i = 1000; i > 0; i /= 10) {
        digit = (num / i + modifier) % 10;
        num = num % i + digit * 10;
        modifier *= 10;
    }
    return num;
}
int encrypt(int num) {
    return transform((num % 1000 / 100) * 1000 + (num % 100 / 10) * 100 +
(num % 10) * 10 + num / 1000, 7);
}
int decrypt(int num) {
    return transform((num % 1000 / 100) * 1000 + (num % 10) * 100 + (num %
100 / 10) * 10 + num / 1000, 3);
}
int main() {
  int original, encrypted, decrypted;
cout << "Enter a four-digit integer to encrypt: ";
cin >> original;

cout << "Encrypted number: " << encrypted << endl;
    encrypted = encrypt(original);
```

```
cout << "\nEnter a four-digit encrypted integer to decrypt: ";
 cin >> encrypted;


    decrypted = decrypt(encrypted);
cout << "Decrypted number: " << decrypted << endl;
 return 0;
}
```

CHAPTER NO 5

5.11)

The code for the smallest integers is

```
#include <iostream>

using namespace std;

int main()
{
    int count;
    int min, num;

    cout << "Enter number of integers: " << endl;
    cin >> count;

    for(int i = 1; i <= count; i++){
        cout << "Enter number: " << endl;
        cin >> num;
        if(i == 1){
            min = num;
        }
        if(num < min){
            min = num;
        }
    }

    cout << "The smallest number is: " << min << endl;

    return 0;
}
```

5.12)

The following program is used to calculate the product of the even integers from 2 to 10.

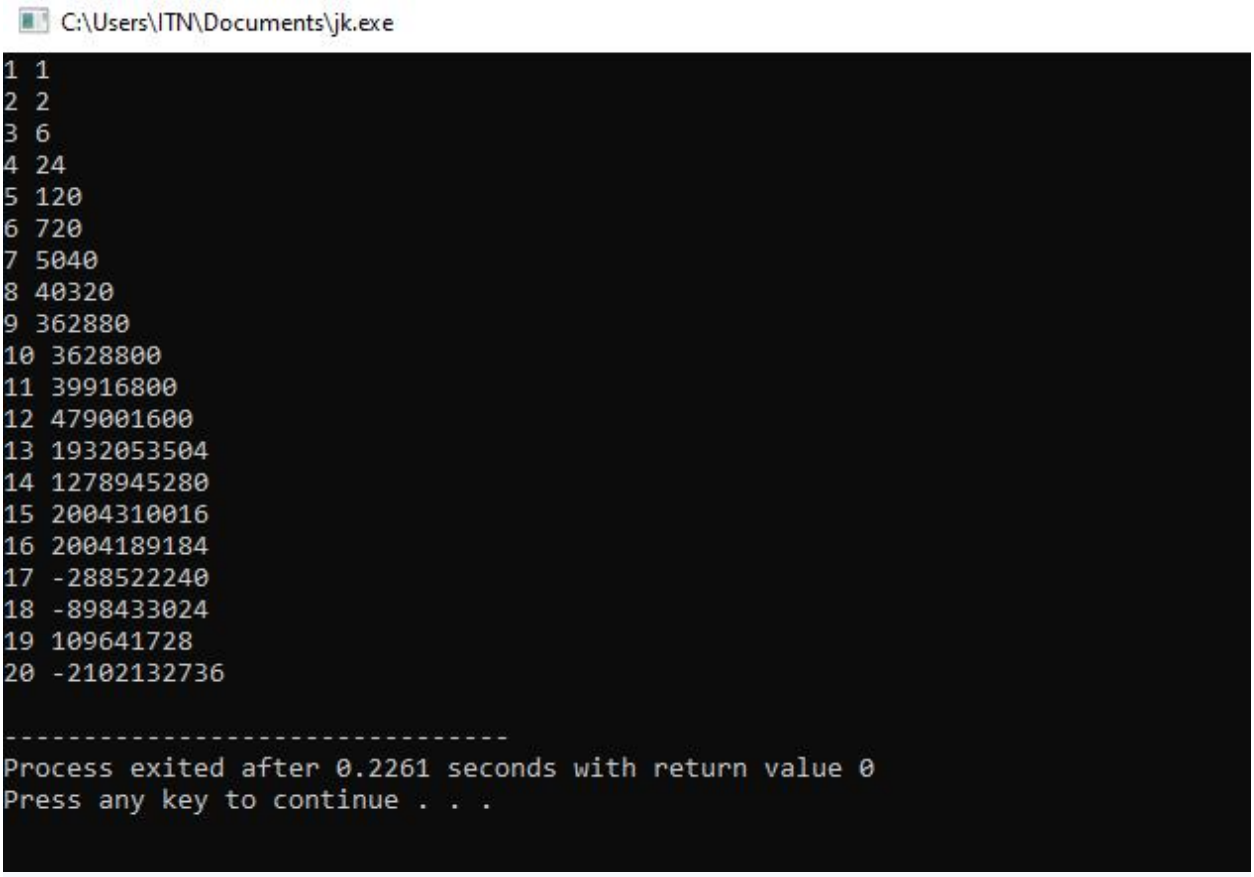
```
#include <iostream>

using namespace std;

int main()

{

  int product = 1;

      for(int i=2; i <= 10; i+=2){

        product *= i;

    }

      cout << "The product of the even integers between 2 and 10 is: " <<
product << endl;

   return 0;

}
```

5.13)

The following program calculates the factorials and prints them in tabular form.

```
#include <iostream>

using namespace std;

int main()
{
    long fact = 1;

    for(int i=1; i <= 20; i++){
        fact *= i;
        cout << i << " " << fact << endl;
    }


    return 0;
}
```

OUTPUT:



Result: The factorial of 100100 is a number with more than 150150 digits and it can not be stored in a C++ variable.

5.15

Write a program that uses for statements to print the following patterns separately,one below the other. Use for loops to generate the patterns.All asterisks(*) should be printed by a single statement of the form cout<<'*'; (this causes the asterisks to print side by side).[Hint:The last two patterns require that each line begin with an appropriate number of blanks.Extra credit:Combine your code from the four separate problem into a single program that prints all four patterns side by side by making clever use of nested for loops.]

a)

```
#include<iostream>

using namespace std;

main()

{
```

```
for(int u=1; (u<=10); u=u+1)
{for(int i=1; (i<=u);i=i+1)
cout<<"*";
cout<<endl;
}
return 0;
}
```

b)

```
#include<iostream>
using namespace std;
main()
{
      for(int u=10; (u>=1); u=u-1)
      {for(int i=1; (i<=u);i=i+1)
      cout<<"*";
      cout<<endl;
      }
      return 0;
}
```

C)

```
#Include<Iostream>
Using Namespace Std;
Main ()
{

Int i = 1;
 While (i <= 10) {
 Int j = 1;
 While (j <= i - 1) {
```

```
Cout << ' ';
 j++;
}
 Int k = 1;
 While (k <= 11 - i) {
 Cout << '*';
 k++;
 }
 Cout << '\n';
 i++;
 }
 Return 0;
}
```

d)

```
#Include<Iostream>
Using Namespace Std;
Main ()

 {

   Int i=1;
   While (i<=10) {
   Int j=1;
   While (j<=10-i) {
   Cout<<' ';
   j++;
}
   Int k = 1;
   While (k <= i) {
```

```
  Cout<<'*';
  k++;
}
  Cout<<'\n';
  i++;
}
  Return 0;
}
```

5.16

(BarChart)One interesting application of computers is drawingg graph sand bar charts.Write a program that reads five numbers(eachbetween1and30).Assume that the user enters only valid values.For each number that is read,yourprogram should print a line containing that number of adjacent asterisks.For example,if your program reads the number7,it should print*****.**

```
#include<iostream>
usingnamespacestd;
intmain(){
constintnumberOfValues=5;
intvalues[numberOfValues];
cout<<"Enterfivenumbersbetween1and30:"<<endl;
for(inti=0;i<numberOfValues;++i){
do{
cout<<"Number"<<i+1<<":";
cin>>values[i];
if(values[i]<1||values[i]>30){
cout<<"Pleaseenteranumberbetween1and30."<<endl;
}
}while(values[i]<1||values[i]>30);
}
```

```
cout<<"\nBarChart:"<<endl;
for(inti=0;i<numberOfValues;++i){
cout<<i+1<<":";
for(intj=0;j<values[i];++j){
cout<<'*';
}
cout<<endl;
}
return0;
}
```

5.17

```
#include<iostream>
using namespace std;
main ()
 {
    // Product prices
    const double prices[] = {2.98, 4.50, 9.98, 4.49, 6.87};

    int productNumber;
    int quantitySold;
    double totalRetailValue = 0.0;

    // Sentinel-controlled loop
    while (true) {
        // Input product number
        cout << "Enter product number (1-5, or 0 to exit): ";
```

```
        cin >> productNumber;

        // Check for sentinel value
        if (productNumber == 0) {
            break;
        }

        // Input quantity sold
        cout << "Enter quantity sold: ";
        cin >> quantitySold;

        // Validate product number
        if (productNumber >= 1 && productNumber <= 5) {
            // Calculate and display retail value
            double retailValue = prices[productNumber - 1] *
quantitySold;
            cout << "Retail value for product " << productNumber << ":
$" << retailValue << endl;

            // Add to total retail value
            totalRetailValue += retailValue;
        } else {
            cout << "Invalid product number. Please enter a number
between 1 and 5." << endl;
        }
    }

    // Display total retail value
    cout << "Total retail value of all products sold: $" <<
totalRetailValue << endl;

    return 0;}
```

5.18

a)

```
#include<iostream>
using namespace std;
main()
{
     int x=7;
     int y=5;
     int z=3;
     cout<<(x==7&&y==5)<<endl;
     return 0;
}
```

b)

```
#include<iostream>
using namespace std;
main()
{
     int x=7;
     int y=5;
     int z=3;
     cout<<(x==7||y==3)<<endl;
     return 0;
}
```

c)

```
#include<iostream>
using namespace std;
main()
{
     int x=7;
     int y=5;
```

```
    int z=3;
    cout<<(x==7&&y==3)<<endl;
    return 0;
}
```

d)
```
#include<iostream>
using namespace std;
main()
{
    int x=7;
    int y=5;
    int z=3;
    cout<<(z==7||y==3)<<endl;
    return 0;
}
```

e)
```
#include<iostream>
using namespace std;
main()
{
    int x=7;
    int y=5;
    int z=3;
    cout<<(x-2==5&&y+4==9)<<endl;
    return 0;
}
```

f)#include<iostream>

```
using namespace std;
main()
{
     int x=7;
     int y=5;
     int z=3;
     cout<<(x>=7||y<=5)<<endl;
     return 0;
}
```

g)#include<iostream>

```
using namespace std;
main()
{
     int x=7;
     int y=5;
     int z=3;
     cout<<(y!=6&&z!=3)<<endl;
     return 0;
}
```

h)#include<iostream>

```
using namespace std;
main()
{
     int x=7;
     int y=5;
```

```
     int z=3;
     cout<<(y!=5||z!=6)<<endl;
     return 0;
}
```

i)

```
#include<iostream>
using namespace std;
main()
{
     int x=7;
     int y=5;
     int z=3;
     cout<<(!(y-z-2))<<"\t"<<(!(y>z))<<endl;
     return 0;
}
```

5.19

```
#include <iostream>

using namespace std;

int main() {
 double pi{ 0.0 };
 cout << "Step\tPi\n" << endl;
for ( int i{ 1 }; i <= 200000; ++i ) {
 double term{ 4.0 / ( i * 2 - 1 ) };
 i % 2 ? pi += term : pi -= term;
 cout << i << '\t' << pi << endl; // 3.14159 - step 130658
   }
  return 0;
}
```

5.20

```
#include<iostream>

using namespace std;

main()

{

      cout<<"Pythagorean triples with all sides no longer than
500"<<endl;

for(int side1=1; side1<=500; ++side1)


for(int side2=1; side2<=500; ++side2)


for(int hypotenuse=1; hypotenuse<=500; ++hypotenuse)
      if((side1*side1)+side2*side2 == hypotenuse*hypotenuse)

      cout<<"("<<side1<<","<<side2<<","<<hypotenuse<<")"<<endl;

return 0;
```

}5.21

```
#include <iostream>

using namespace std;

main ()

 {

    // Loop for each row

    for (int i = 1; i<= 5; ++i) {


   // First Triangle

        for (int j = 1; j <= i; ++j) {

            cout << "*";

        }

        // Spaces between triangles

        for (int j = 1; j <= 5 - i; ++j) {

            cout << " ";
```

```
}
// Second Triangle
for (int j = 1; j <= 5 - i; ++j) {
    cout << " ";
}
for (int j = 1; j <= i; ++j) {
    cout << "*";
}

// Spaces between triangles
for (int j = 1; j <= 5; ++j) {
    cout << " ";
}

// Third Triangle
for (int j = 1; j <= 5 - i; ++j) {
    cout << " ";
}
for (int j = 1; j <= i; ++j) {
    cout << "*";
}

// Spaces between triangles
for (int j = 1; j <= i - 1; ++j) {
    std::cout << " ";
}

// Fourth Triangle
for (int j = 1; j <= i; ++j) {
    cout << "*";
```

```
        }

        // Move to the next line for the next row
        cout << std::endl;
    }

    return 0;
}
```

5.23

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

int main() {
 //sec1
 for(int i=1; i<=9; ++i)
 cout<<"#";
 cout<<endl;
 //sec2
 for(int r=1; r<=4; ++r)
 {
 cout<<"#";
  for(int sp=1; sp<r; ++sp)
 cout<<" ";

 for(int st=7; st>=2*r-1; --st)
 cout<<"*";
```

```
 for(int sp=1; sp<r; ++sp)
cout<<" ";
cout<<"#";
  cout<<endl;
  }
//sec3
  for(int r=4-1; r>=1; --r)
  {
 cout<<"#";
    for(int sp=1; sp<r; ++sp)
            cout<<" ";
            for(int st=7; st>=2*r-1; --st)
            cout<<"*";
            for(int sp=1; sp<r; ++sp)
            cout<<" ";
                        cout<<"#";
            cout<<endl;
  }   //sec4
  for(int i=1; i<=9; ++i)
    cout<<"#";
    cout<<endl;
}
```

5.24

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

```
int main() {
      int h;
      cout<<"enter the odd height of egg timer in range from 1 to 29: ";
      cin>>h;

  //sec1
  for(int i=1; i<=h; ++i)
  cout<<"#";
  cout<<endl;
  //sec2
  for(int r=1; r<=h/2; ++r)
  {
      cout<<"#";
    for(int sp=1; sp<r; ++sp)
      cout<<" ";

      for(int st=h-2; st>=2*r-1; --st)
      cout<<"*";

      for(int sp=1; sp<r; ++sp)
      cout<<" ";
            cout<<"#";
      cout<<endl;
  }
      //sec3
    for(int r=h/2-1; r>=1; --r)
  {
      cout<<"#";
    for(int sp=1; sp<r; ++sp)
      cout<<" ";
```

```
      for(int st=h-2; st>=2*r-1; --st)
      cout<<"*";

      for(int sp=1; sp<r; ++sp)
      cout<<" ";
            cout<<"#";
      cout<<endl;
  }  //sec4
  for(int i=1; i<=h; ++i)
     cout<<"#";
     cout<<endl;
}
```

5.25

```
#include <iostream>
Using namespace std;
int main() {
  bool exitLoop = false;

  for (int i = 0; i < 10; ++i) {
    // Body of the loop

    if (condition) {
      // Code that would be inside the loop after encountering break

      exitLoop = true;
    }

    // Additional condition to check for early exit
```

```
    if (exitLoop) {
        break;
    }

    // Rest of the loop logic
  }

  // Code outside the loop

  return 0;
}
```

(5.26)

What does the following program segment do?

```
for (int x {1}  ;  x<=  2;   x++  )  {
   for    (int* y {5} ;  y<=1  ;  y- -  )  {
       for    (int   z {1};  z<=  y;  z++  )   {
              Cout    << z *   x << "*";
}
  Cout  <<  endl;
  }
}
#include <iostream>
using namespace std;

int main() {
   for (int x = 1; x <= 2; x++) {
```

```
      for (int y = 5; y >= 1; y--) {
         for (int z = 1; z <= y; z++) {
            if (z <= y) {
               cout << z * x << " ";
            } else {
               cout << endl;
            }
         }
      }
   }
return 0;
}
```

(5.27)

Describe in general how you'd remove any continue statement from a loop in a program and replace it with some structured equivalent. Use the technique you develop here to remove the continue statement from the program.

```
#include<iostream>
Using namespace std;
Int main() {
    for (int i = 0; i < 10; ++i) {
        if (i != 3 && i != 7) {
cout << i << " ";
        }
}
    return 0;
```

```
}
```

(5.28)

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

void printVerse(int day) {
    switch (day) {
        case 1:
            cout << "first";
            break;
        case 2:
            cout << "second";
            break;
        case 3:
            cout << "third";
            break;
        case 4:
            cout << "fourth";
            break;
        case 5:
            cout << "fifth";
            break;
        case 6:
            cout << "sixth";
            break;
        case 7:
            cout << "seventh";
            break;
        case 8:
            cout << "eighth";
            break;
```

```
    case 9:
        cout << "ninth";
        break;
    case 10:
        cout << "tenth";
        break;
    case 11:
        cout << "eleventh";
        break;
    case 12:
        cout << "twelfth";
        break;
    default:
        cout << "Invalid day";
}
cout << " day of Christmas, my true love sent to me:" << endl;

switch (day) {
    case 12:
        cout << "Twelve drummers drumming," << endl;
    case 11:
        cout << "Eleven pipers piping," << endl;
    case 10:
        cout << "Ten lords a-leaping," << endl;
    case 9:
        cout << "Nine ladies dancing," << endl;
    case 8:
        cout << "Eight maids a-milking," << endl;
    case 7:
        cout << "Seven swans a-swimming," << endl;
    case 6:
        cout << "Six geese a-laying," << endl;
```

```
        case 5:
            cout << "Five golden rings," << endl;
        case 4:
            cout << "Four calling birds," << endl;
        case 3:
            cout << "Three French hens," << endl;
        case 2:
            cout << "Two turtle doves," << endl;
        case 1:
            if (day != 1)
                cout << "And ";
            cout << "A partridge in a pear tree." << endl;
            break;
        default:
            cout << "Invalid day";
    }
    cout << endl;
}

int main() {
    for (int i = 1; i <= 12; ++i) {
        printVerse(i);
    }
    return 0;
}
```

5.29

```
#include<iostream>
using namespace std;
main( )
{
double principle=24.0;
```

```
int year=1626;
int current_year=2023;
int years=current_year- year;
for (double rate=5.0; rate<=10.0;rate+=0.1)
{
      double amount=principle*(1.0+rate/100,years);
      cout<<"after"<<years<<"years at the interest rate
of"<<rate<<"%$"<<principle<<"becomes$"<<amount<<endl;
}
cout<<endl;
return 0;}
```

5.30

```
#include <iostream>
using namespace std;
class DollarAmount {
private:
   int dollars;
   int cents;
   int amount;

public:
   DollarAmount(int dollars, int cents) {
      this->dollars = dollars;
      this->cents = cents;
      this->amount = dollars * 100 + cents;
   }
   void displayAmount() {
      cout << "Total amount in pennies: " << amount << " pennies\n";
   }
```

```
};

int main() {

   DollarAmount myAmount(5, 75);
   myAmount.displayAmount();

   return 0;}
```

5.31

```
}#include <iostream>
using namespace std;
int main()
 {
   int dollars, cents, amount, divisor;
   divisor = 3;
   for (dollars = 1; dollars <= 10; ++dollars)
 {
      for (cents = 0; cents < 100; cents += 10)
{
         amount = dollars * 100 + cents;
if (divisor != 0)
 {
            amount = (amount + divisor / 2) / divisor;
         }

cout << "For " << dollars << " dollars and " << cents << " cents, the total amount in pennies after division by " << divisor << " is: " << amount << endl;
      }
```

5.32

```
#include <iostream>
#include <cmath>
using namespace std;
class DollarAmount {
private:
int dollars;
int cents;
public:
DollarAmount(int dol, int cen) : dollars(dol), cents(cen) {}
void addInterest(double rate) {
double interest = getAmount() * rate;
interest = round(interest);
cents += static_cast<int>(interest);
dollars += cents / 100;
cents %= 100;
}
int getAmount() const {
return dollars * 100 + cents;
}
void display() const {
cout << "Amount: $" << dollars << "." << cents << endl;
}
};
 int main() {
int dollars;
int cents;
double interestRate;
```

```
cout << "Enter the initial amount in dollars: ";
cin >> dollars;
cout << "Enter the initial amount in cents: ";
cin >> cents;
cout << "Enter the interest rate: ";
cin >> interestRate;
DollarAmount amount(dollars, cents);
cout << "Initial ";
amount.display();
amount.addInterest(interestRate);
cout << "After adding interest with Banker's rounding: ";
amount.display();
return 0;
   return 0;
}
```

5.33

```
#include<iostream>
using namespace std;
class DollarAmount {
private:
int dollars;
int cents;
public:
DollarAmount(int dollars = 0, int cents = 0) {
setDollars(dollars);
setCents(cents);
}
void setDollars(int dollars) {
this->dollars = dollars;
```

```
}
void setCents(int cents) {
if (cents >= 0 && cents <= 99) {
this->cents = cents;
} else {
cout << "Invalid cents value. Setting cents to 0." << endl;
this->cents = 0;
}
}
int getDollars() const {
return dollars;
 int getCents() const {
return cents;
}
void addInterest(double rate) {
int totalCents = dollars * 100 + cents;
totalCents = (int)(totalCents * (1 + rate) + 0.5);
dollars = totalCents / 100;
cents = totalCents % 100;
}
void display() const {
cout << "Amount: $" << dollars <<"."<<cents<<endl;
}
};
int main() {
DollarAmount amount(100, 50);
amount.display();
for (int i = 0; i < 5; i++) {
amount.addInterest(0.05);
amount.display();
```

```
}
return 0;
}
```

5.35

```
#include <iostream>
#include <iomanip>

double calculateInterestRate(int principal, int time) {
    double timeDecimal = static_cast<double>(time) / 100.0;

    double interestRate = principal * timeDecimal;

    return interestRate;
}
int main() {
    int principal, time;

    std::cout << "Enter the principal amount: ";
    std::cin >> principal;

    std::cout << "Enter the time in years: ";
    std::cin >> time;

        double interestRate = calculateInterestRate(principal, time);

       if (time < 1000) {
        std::cout << std::fixed << std::setprecision(1);
        std::cout << "The interest rate based on " << principal << " and "
<< time << " is: " << interestRate << "%" << std::endl;
    } else {
        std::cout << std::fixed << std::setprecision(3);
```

```
        std::cout << "The interest rate based on " << principal << " and "
<< time << " is: " << interestRate << "%" << std::endl;
    }

    return 0;
}
```

Question no 5.36

```
#include <iostream>
#include <iomanip>
int main() {
 double double_variable = 123.02;
    for (int precision = 1; precision <= 17; precision++) {
        std::cout << "Precision: " << precision << ", Value: " <<
std::fixed << std::setprecision(precision) << double_variable << std::endl;
    }

    return 0;
}
```

Question no 5.37

```
#include <iostream>
#include <vector>
#include <algorithm>
void displayQuestion(int questionNumber, const std::string& question,
const std::vector<std::string>& choices) {
    std::cout << "Question " << questionNumber << ": " << question <<
std::endl;

    for (size_t i = 0; i < choices.size(); ++i) {
        std::cout << i + 1 << ". " << choices[i] << std::endl;
    }

    std::cout << "Enter your choice (1-4): ";
```

```
}
int evaluateQuiz(const std::vector<int>& userAnswers, const
std::vector<int>& correctAnswers) {
    int correctCount = 0;

    for (size_t i = 0; i < userAnswers.size(); ++i) {
        if (userAnswers[i] == correctAnswers[i]) {
            ++correctCount;
        }
    }

    return correctCount;
}

int main() {
        std::vector<std::string> questions = {
        "What is the primary greenhouse gas responsible for global
warming?",
        "Which of the following human activities contributes to greenhouse
gas emissions?",
        "What is the main argument of global warming skeptics?",
        "What is the role of deforestation in global warming?",
        "How does the Intergovernmental Panel on Climate Change (IPCC)
contribute to the global warming debate?"
    };

    std::vector<std::vector<std::string>> choices = {
        {"Carbon Dioxide (CO2)", "Methane (CH4)", "Nitrous Oxide (N2O)",
"Water Vapor (H2O)"},
        {"Burning fossil fuels", "Eating meat", "Driving cars", "All of
the above"},
        {"Human activities are not significant", "Climate change is
natural", "Data is unreliable", "None of the above"},
```

```
        {"Increases global warming", "Decreases global warming", "Has no
effect", "Not related to global warming"},
        {"Researching climate change", "Denying climate change",
"Promoting climate change", "No involvement"}
    };
    std::vector<int> correctAnswers = {1, 4, 2, 1, 1};
    std::vector<int> userAnswers;
    for (size_t i = 0; i < questions.size(); ++i) {
        displayQuestion(i + 1, questions[i], choices[i]);
        int userChoice;
        std::cin >> userChoice;

          userChoice = std::max(1, std::min(4, userChoice));
        userAnswers.push_back(userChoice);
    }
    // Evaluate quiz and display result
    int correctCount = evaluateQuiz(userAnswers, correctAnswers);

    std::cout << "\nResult: ";
    if (correctCount == 5) {
        std::cout << "Excellent!";
    } else if (correctCount == 4) {
        std::cout << "Very good.";
    } else {
        std::cout << "Time to brush up on your knowledge of global
warming.";
    }

    std::cout << "\n\nSources:\n";
    std::cout << "- An Inconvenient Truth:
    std::cout << "- Intergovernmental Panel on Climate Change:    return 0;
}
```

5.38

(Tax Plan Alternatives; The “Fair Tax”) There are many (often controversial) proposals to make taxation “fairer”. Check out the Fair Tax initiative in the United States at ww.fairtax.org. Research how the proposed Fair Tax works. One suggestion is to eliminate income taxes and most other taxes in favor of a 23% consumption tax on all products and services that u buy. Some Fair-Tax opponents question the 23% figure and say that because of the way the tax is calculated, it would be more accurate to say that rate is 30%___check this carefully. Write a program that prompts the user to enter expenses in various expense categories the have (e.eg., housing, food, clothing, transportation, education, health care, vacations), then prints the estimated FairTax that person would play.

```
#include <iostream>
#include <map>
int main()
 {
    std::map<std::string, double> expenseCategories;
    double totalExpenses = 0.0;
    std::cout << "Enter your expenses in various categories. Type 'done' to finish.\n";
    while (true) {
        std::string category;
        double expense;

        std::cout << "Enter category (or 'done' to finish): ";
        std::cin >> category;

        if (category == "done") {
            break;
        }

        std::cout << "Enter expense for " << category << ": ";
```

```
        std::cin >> expense;


        expenseCategories[category] = expense;
        totalExpenses += expense;
    }
    const double fairTaxRate = 0.23;
    double fairTax = totalExpenses * fairTaxRate;
    std::cout << "\nEstimated Fair Tax: $" << fairTax << std::endl;
          return 0;
}
```

5.39.

(The Global Economy) The United Nations recognizes 180 world currencies as legal tender. Investigate a few of these by looking at the list available at https://en.wikipedia.org/wiki/List_of_ciculating_currencies Each currency will have a dollar exchange rate. Investigate the cost of bread in a few of these currencies. Write a program that allows the user to enter a unit of currency and the associated dollar exchange rate for that currency. The user should be allowed to enter a maximum of 5 different currency-exchange pairs. If they would like to stop entering currencies, they may enter the character'x'. The user should then be prompted to enter a dollar amount and, for each currency entered, the equivalent value should be shown. The user should enter -1 to stop calculating equivalent currency values.

```
#include <iostream>
#include <string>
#include <iomanip>
int main()
{
    const int MAX_CURRENCIES = 5;
    std::string currencies[MAX_CURRENCIES];
    double exchangeRates[MAX_CURRENCIES];
    int numCurrencies = 0;
    std::cout << "Enter up to " << MAX_CURRENCIES << " currency-
exchange rate pairs (or 'x' to stop):\n";
```

```
    for (int i = 0; i < MAX_CURRENCIES; ++i) {
        std::cout << "Enter currency code (e.g., USD): ";
        std::cin >> currencies[i];

        if (currencies[i] == "x") {
            break;
        }

        std::cout << "Enter exchange rate for " << currencies[i] << "
to USD: ";
        std::cin >> exchangeRates[i];

        ++numCurrencies;
    }
    double dollarAmount;
    std::cout << "Enter a dollar amount (or -1 to stop): ";
    std::cin >> dollarAmount;

    while (dollarAmount != -1) {
        std::cout << "Equivalent values:\n";
        for (int i = 0; i < numCurrencies; ++i) {
            double equivalentValue = dollarAmount * exchangeRates[i];
            std::cout << currencies[i] << ": " << std::fixed <<
std::setprecision(2) << equivalentValue << '\n';
        }
        std::cout << "\nEnter a dollar amount (or -1 to stop): ";
        std::cin >> dollarAmount;
    }

    std::cout << "Program terminated.\n";
```

```
    return 0;
}
```