#7 Half Number

    int number;

    cout << "Please enter the first number  ? \n";

    cin >> number;

    cout << "Half of " << number << " is " << number / 2;

#9 sum of 3 numbers

    short a, b, c;

    cout << "Please enter the first number  A ? \n";

    cin >> a;

    cout << "Please enter the first number   B? \n";

    cin >> b;

    cout << "Please enter the first number  C? \n";

    cin >> c;

    int sum = a + b + c;

    cout << "Output is : " << sum;

#10 avg of 3 marks

    short a, b, c;

    cout << "Please enter the first mark1 ? \n";

    cin >> a;

    cout << "Please enter the first mark2? \n";

    cin >> b;

    cout << "Please enter the first mark3? \n";

    cin >> c;

    int avg = (a + b + c) / 3;

    cout << "Output is : " << avg;

}

#14 swap numbers

    short a, b;

    cout << "Please enter the first number  A ? \n";

    cin >> a;

    cout << "Please enter the first number   B? \n";

    cin >> b;

    cout << "Output : \n";

    cout << a << "\n"

         << b << "\n \n";

    short temp = a;

    a = b;

    b = temp;

    cout << a << "\n"

         << b << "\n";

#15 Rectangle area

   short width, length;

    cout << "Please enter  width ? \n";

    cin >> width;

    cout << "Please enter  length? \n";

    cin >> length;

    short rectangleArea = width \* length;

    cout << "Rectangle area is :" << rectangleArea;

#17 Triangle area

 short base, height;

    cout << "Please enter  base ? \n";

    cin >> base;

    cout << "Please enter  height? \n";

    cin >> height;

    float triangleArea = 0.5f \* base \* height; // f==>float

    cout << "Triangle area is : " << triangleArea;

#19 Circle Area through diameter

    float diameter;

    cout << "Please enter  circle  diameter ? \n";

    cin >> diameter;

    float radius = diameter / 2;

    const float PI = 3.14;

    float circleArea = PI \* radius \* radius; // f==>float

    cout << "circle area is : " << circleArea;

#20 circle area inscribed in a square

    float squareSideLength;

    cout << "Please enter  square side length ? \n";

    cin >> squareSideLength;

    const float PI = 3.14;

    float circleArea = (PI \* squareSideLength \* squareSideLength) / 4;

    cout << "circle area is : " << circleArea;

#21 circle area along the circumference

float circleCircumference;

    cout << "Please enter  circle circumference ? \n";

    cin >> circleCircumference;

    const float PI = 3.14;

    float circleArea = (circleCircumference \* circleCircumference) / (4 \* PI);

    cout << "circle area is : " << circleArea;

#22 circle area inscribed in an isosceles triangle

  float triangleequalSides, trianglebase;

    cout << "Please enter triangle sides length ,1-triangleequalSides 2-  trianglebase? \n";

    cin >> triangleequalSides;

    cin >> trianglebase;

    const double PI = 3.141592653589793;

    float circleArea = (PI \* trianglebase \* trianglebase / 4) \* ((2 \* triangleequalSides - trianglebase) / (2 \* triangleequalSides + trianglebase));

    cout << fixed << setprecision(3); // 4 أرقام بعد الفاصلة

    cout << "circle area is : " << circleArea;

#31 Power 2,3,4

  float number;

    cout << "Please enter  number : \n";

    cin >> number;

    cout << "power 2: " << number \* number << "\n";

    cout << "power 3: " << number \* number \* number << "\n";

    cout << "power 4: " << number \* number \* number \* number << "\n";

#35 Piggy Bank Calculator

 float number;

    short penny, nickel, dime, quarter, dollar;

    const short Penny = 1, Nickel = 5, Dime = 10, Quarter = 25, Dollar = 100;

    cout << "Please enter  pennies  : \n";

    cin >> penny;

    cout << "Please enter  nickels  : \n";

    cin >> nickel;

    cout << "Please enter  dimes  : \n";

    cin >> dime;

    cout << "Please enter  quarter  : \n";

    cin >> quarter;

    cout << "Please enter  dollar  : \n";

    cin >> dollar;

    float totalPennies = penny \* Penny + dime \* Dime + nickel \* Nickel + quarter \* Quarter + dollar \* Dollar;

    float totalDollars = totalPennies / 100;

    cout << "Total Pennies : " << totalPennies << "\n";

    cout << "TotalDollars: " << totalDollars << "\n";

#39 Pay reminder

float number;

    short cashPaid, totalBill;

    cout << "Please enter  cash Paid  : \n";

    cin >> cashPaid;

    cout << "Please enter  Total Bill   : \n";

    cin >> totalBill;

    float reminder = cashPaid - totalBill;

    cout << "reminder : " << reminder << "\n";

#40 Service fee and Sales Tax

 float billValue;

    cout << "Please enter  Bill Value   : \n";

    cin >> billValue;

    billValue += (billValue \* 0.1);

    billValue += (billValue \* 0.16);

    cout << "After Service and tax fee , bill will be : " << billValue << "\n";

#42 Task Duration in seconds

   float numOfDays, numOfHours, numOfMinutes, numOfSeconds;

    cout << "Please enter number of days    : \n";

    cin >> numOfDays;

    cout << "Please enter number of hours    : \n";

    cin >> numOfHours;

    cout << "Please enter number of minutes    : \n";

    cin >> numOfMinutes;

    cout << "Please enter number of seconds    : \n";

    cin >> numOfSeconds;

    float totalSecondsFromDays = numOfDays \* 24 \* 60 \* 60;

    float totalSecondsFromHours = numOfHours \* 60 \* 60;

    float totalSecondsFromMinutes = numOfMinutes \* 60;

    float totalSeconds = totalSecondsFromDays + totalSecondsFromHours + totalSecondsFromMinutes + numOfSeconds;

    cout << "Total Seconds : " << totalSeconds;

#43 seconds to Days ,Hours, Minutes ,Seconds

    const float secondsPerDay = 24 \* 60 \* 60;

    const float secondsPerHour = 60 \* 60;

    const float secondsPerMinute = 60;

    float totalSeconds;

    float reminder;

    float numOfSeconds;

    cout << "Please enter totalSeconds   : \n ";

    cin >> totalSeconds;

    float numOfDays = floor(totalSeconds / secondsPerDay);

    reminder = fmod(totalSeconds, secondsPerDay);

    float numOfHour = floor(reminder / secondsPerHour);

    reminder = fmod(reminder, secondsPerHour);

    float numOfMinutes = floor(reminder / secondsPerMinute);

    reminder = fmod(reminder, secondsPerMinute);

    numOfSeconds = reminder;

    cout << numOfDays << " : " << numOfHour << ": " << numOfMinutes << ": " << numOfSeconds;

#47 Loan installment months

    float loanAmount, monthlyPayment;

    cout << "Please enter Loan Amounts   : \n ";

    cin >> loanAmount;

    cout << "Please enter  monthly payment   : \n ";

    cin >> monthlyPayment;

    float monthToSettleLoan = loanAmount / monthlyPayment;

    cout << "Total Months : " << monthToSettleLoan << endl;

#48Monthly loan installment

    float loanAmount, numOfMonths;

    cout << "Please enter Loan Amounts   : \n ";

    cin >> loanAmount;

    cout << "Please enter  num of months   : \n ";

    cin >> numOfMonths;

    float loanPerMonth = loanAmount / numOfMonths;

    cout << "Loan Per Month : " << loanPerMonth << endl;

Use Math

#16 Rectangle Area Through Diagonal and side area =🡺using pow

  double rectangleSide;

     double rectangleDiagonal;

     cout << "Please enter rectangleSide   : \n ";

     cin >> rectangleSide;

     cout << "Please enter rectangleDiagonal  : \n ";

     cin >> rectangleDiagonal;

     double area = rectangleSide \* sqrt(pow(rectangleDiagonal, 2) - pow(rectangleSide, 2));

     cout << "Area of Rectangle : " << area << endl;

#18 circle area =🡺using ceil

   double circleRadius;

     const float PI = 3.144444;

     cout << "Please enter radius   : \n ";

     cin >> circleRadius;

     double area = ceil(PI \* pow(circleRadius, 2));

     cout << "Area of circle  : " << area << endl;

#19 circle area through diameter =🡺 using ceil

 double circleDiameter;

     const float PI = 3.144444;

     cout << "Please enter diameter   : \n ";

     cin >> circleDiameter;

     double area = ceil(PI \* pow(circleDiameter / 2, 2));

     cout << "Area of circle  : " << area << endl;

#20circle area inscribed in the square =🡺 using ceil

double squareSide;

     double rectangleDiagonal;

     const float PI = 3.144444;

     cout << "Please enter square side   : \n ";

     cin >> squareSide;

     double area = ceil(PI \* pow(squareSide / 2, 2));

     cout << "Area of circle inscribed in the square is  : " << area << endl;

#21 circle area along the circumference 🡺using floor

  double circleCircumference;

     const float PI = 3.144444;

     cout << "Please enter circumference   : \n ";

     cin >> circleCircumference;

     double area = floor(pow(circleCircumference, 2) / (4 \* PI));

     cout << "Area of circle  : " << area << endl;

#22 circle area inscribed in an isosceles 🡺using floor

  double triangleSides;

     double triangleBase;

     const float PI = 3.144444;

     cout << "Please enter triangleSides   : \n ";

     cin >> triangleSides;

     cout << "Please enter triangleBase   : \n ";

     cin >> triangleBase;

     double area = PI \* pow(triangleBase, 2) / 4 \* ((2 \* triangleSides - triangleBase) / (2 \* triangleSides + triangleBase));

     area = floor(area);

     cout << "Area of circle  : " << area << endl;

#23 circle area **circle Described Around an Arbitrary Triangle =🡺using round**

     double triangleSideA, triangleSideB, triangleSideC;

     double triangleBase;

     const float PI = 3.144444;

     cout << "Please enter triangleSides A,B,C  in order   : \n ";

     cin >> triangleSideA;

     cin >> triangleSideB;

     cin >> triangleSideC;

     double HalfOftriangleCircumference = (triangleSideA + triangleSideB + triangleSideC) / 2;

     double radius = triangleSideA \* triangleSideB \* triangleSideC / (4 \* sqrt(HalfOftriangleCircumference \* (HalfOftriangleCircumference - triangleSideA) \* (HalfOftriangleCircumference - triangleSideB) \* (HalfOftriangleCircumference - triangleSideC)));

     double area = PI \* pow(radius, 2);

     area = round(area);

     cout << "Area of circle  : " << area << endl;

#31 Pow of 2,3,4 ==🡺 using round\

     double number;

     cout << "Please enter number   : \n ";

     cin >> number;

     cout << number << "pow 2 is :" << round(pow(number, 2)) << endl;

     cout << number << "pow 3 is :" << round(pow(number, 3)) << endl;

     cout << number << "pow 4 is :" << round(pow(number, 4)) << endl;

#32 Pow of M =🡺 using round

 double number;

     double power;

     double result = 1;

     cout << "Please enter number   : \n ";

     cin >> number;

     cout << "Please enter  power    : \n ";

     cin >> power;

     for (int i = 0; i < power; i++)

     {

          result \*= number;

     }

     cout << number << " pow " << power << " is :" << round(result) << endl;

or

     double number;

     double power;

     cout << "Please enter number   : \n ";

     cin >> number;

     cout << "Please enter  power    : \n ";

     cin >> power;

     cout << number << " pow " << power << " is :" << round(pow(number, power)) << endl;

#42 Task duration in seconds =🡺 using round

 float numOfDays, numOfHours, numOfMinutes, numOfSeconds;

     cout << "Please enter number of days    : \n";

     cin >> numOfDays;

     cout << "Please enter number of hours    : \n";

     cin >> numOfHours;

     cout << "Please enter number of minutes    : \n";

     cin >> numOfMinutes;

     cout << "Please enter number of seconds    : \n";

     cin >> numOfSeconds;

     float totalSecondsFromDays = numOfDays \* 24 \* 60 \* 60;

     float totalSecondsFromHours = numOfHours \* 60 \* 60;

     float totalSecondsFromMinutes = numOfMinutes \* 60;

     float totalSeconds = totalSecondsFromDays + totalSecondsFromHours + totalSecondsFromMinutes + numOfSeconds;

     cout << "Total Seconds : " << round(totalSeconds);

#43 seconds to Days ,Hours, Minutes ,Seconds==🡺 floor

    const float secondsPerDay = 24 \* 60 \* 60;

    const float secondsPerHour = 60 \* 60;

    const float secondsPerMinute = 60;

    float totalSeconds;

    float reminder;

    float numOfSeconds;

    cout << "Please enter totalSeconds   : \n ";

    cin >> totalSeconds;

    float numOfDays = floor(totalSeconds / secondsPerDay);

    reminder = fmod(totalSeconds, secondsPerDay);

    float numOfHour = floor(reminder / secondsPerHour);

    reminder = fmod(reminder, secondsPerHour);

    float numOfMinutes = floor(reminder / secondsPerMinute);

    reminder = fmod(reminder, secondsPerMinute);

    numOfSeconds = reminder;

    cout << numOfDays << " : " << numOfHour << ": " << numOfMinutes << ": " << numOfSeconds;

Use Functions

#14

void swapProcedure(short &a, short &b)

{

    short temp = 0;

    temp = a;

    a = b;

    b = temp;

}

int main()

{

    short a, b;

    cout << "Please enter the first number  A ? \n";

    cin >> a;

    cout << "Please enter the first number   B? \n";

    cin >> b;

    cout << "Output : \n";

    cout << a << "\n"

         << b << "\n \n";

    swapProcedure(a, b);

    cout << a << "\n"

         << b << "\n";

    return 0;

}

#15

short rectangleArea(short width, short length)

{

    short rectangleArea = width \* length;

    return rectangleArea;

}

int main()

{

    short a, b;

    short width, length;

    cout << "Please enter  width ? \n";

    cin >> width;

    cout << "Please enter  length? \n";

    cin >> length;

    cout << "Rectangle area is :" << rectangleArea(width, length);

    return 0;

}

#16

double rectangleArea(double rectangleDiagonal, short rectangleSide)

{

    double area = rectangleSide \* sqrt(pow(rectangleDiagonal, 2) - pow(rectangleSide, 2));

    return area;

}

int main()

{

    double rectangleSide;

    double rectangleDiagonal;

    cout << "Please enter rectangleSide   : \n ";

    cin >> rectangleSide;

    cout << "Please enter rectangleDiagonal  : \n ";

    cin >> rectangleDiagonal;

    double area = rectangleArea(rectangleDiagonal, rectangleSide);

    cout << "Area of Rectangle : " << area << endl;

    return 0;

}

#18

double circleArea(double radius)

{

    const float PI = 3.14;

    double area = ceil(PI \* pow(radius, 2));

    return area;

}

int main()

{

    double circleRadius;

    cout << "Please enter radius   : \n ";

    cin >> circleRadius;

    double area = circleArea(circleRadius);

    cout << "Area of circle  : " << area << endl;

    return 0;

}

#19

double circleArea(double diameter)

{

    const float PI = 3.14;

    double radius = diameter / 2;

    double area = PI \* pow(radius, 2);

    return area;

}

int main()

{

    double diameter;

    cout << "Please enter  circle  diameter ? \n";

    cin >> diameter;

    // f==>float

    cout << "circle area is : " << circleArea(diameter);

    return 0;

}

#20

float circleArea(float squareSideLength)

{

    const float PI = 3.14;

    float circleArea = (PI \* squareSideLength \* squareSideLength) / 4;

    return circleArea;

}

int main()

{

    float squareSideLength;

    cout << "Please enter  square side length ? \n";

    cin >> squareSideLength;

    const float PI = 3.14;

    float area = circleArea(squareSideLength);

    cout << "circle area is : " << area;

    return 0;

}

#21

float circleArea(float circleCircumference)

{

    const float PI = 3.14;

    float circleArea = (circleCircumference \* circleCircumference) / (4 \* PI);

    return circleArea;

}

int main()

{

    float circleCircumference;

    cout << "Please enter  circle circumference ? \n";

    cin >> circleCircumference;

    float area = circleArea(circleCircumference);

    cout << "circle area is : " << area;

    return 0;

}

#22

float circleArea(float triangleequalSides, float trianglebase)

{

    const float PI = 3.14;

    float circleArea = (PI \* trianglebase \* trianglebase / 4) \* ((2 \* triangleequalSides - trianglebase) / (2 \* triangleequalSides + trianglebase));

    return circleArea;

}

int main()

{

    float triangleequalSides, trianglebase;

    cout << "Please enter triangle sides length ,1-triangleequalSides 2-  trianglebase? \n";

    cin >> triangleequalSides;

    cin >> trianglebase;

    float area = circleArea(triangleequalSides, trianglebase);

    cout << fixed << area; // 4 أرقام بعد الفاصلة

    cout << "circle area is : " << area;

    return 0;

}

#23

double circleArea(double triangleSideA, double triangleSideB, double triangleSideC)

{

    const float PI = 3.14;

    double HalfOftriangleCircumference = (triangleSideA + triangleSideB + triangleSideC) / 2;

    double radius = triangleSideA \* triangleSideB \* triangleSideC / (4 \* sqrt(HalfOftriangleCircumference \* (HalfOftriangleCircumference - triangleSideA) \* (HalfOftriangleCircumference - triangleSideB) \* (HalfOftriangleCircumference - triangleSideC)));

     double area = PI \* pow(radius, 2);

    return area;

}

int main()

{

     double triangleSideA, triangleSideB, triangleSideC;

     double triangleBase;

     cout << "Please enter triangleSides A,B,C  in order   : \n ";

     cin >> triangleSideA;

     cin >> triangleSideB;

     cin >> triangleSideC;

     double HalfOftriangleCircumference = (triangleSideA + triangleSideB + triangleSideC) / 2;

     double area=circleArea(triangleSideA,triangleSideB,triangleSideC);

     cout << "Area of circle  : " << area << endl;

    return 0;

}

#31

void power234(float number)

{

    cout << "power 2: " << pow(number, 2) << "\n";

    cout << "power 3: " << pow(number, 3) << "\n";

    cout << "power 4: " << pow(number, 4) << "\n";

}

int main()

{

    float number;

    cout << "Please enter  number : \n";

    cin >> number;

    power234(number);

    return 0;

}

#32

double powerOfN(double power, double number)

{

    double result = 1;

    for (int i = 0; i < power; i++)

    {

        result \*= number;

    }

    return result;

}

int main()

{

    double number;

    double power;

    double result = 1;

    cout << "Please enter number   : \n ";

    cin >> number;

    cout << "Please enter  power    : \n ";

    cin >> power;

    result = powerOfN(power, number);

    cout << number << " pow " << power << " is :" << round(result) << endl;

    return 0;

}

#42

float totalSeconds(float numOfDays, float numOfHours, float numOfMinutes, float numOfSeconds)

{

    float totalSecondsFromDays = numOfDays \* 24 \* 60 \* 60;

    float totalSecondsFromHours = numOfHours \* 60 \* 60;

    float totalSecondsFromMinutes = numOfMinutes \* 60;

    float totalSeconds = totalSecondsFromDays + totalSecondsFromHours + totalSecondsFromMinutes + numOfSeconds;

    return totalSeconds;

}

int main()

{

    float numOfDays, numOfHours, numOfMinutes, numOfSeconds;

    cout << "Please enter number of days    : \n";

    cin >> numOfDays;

    cout << "Please enter number of hours    : \n";

    cin >> numOfHours;

    cout << "Please enter number of minutes    : \n";

    cin >> numOfMinutes;

    cout << "Please enter number of seconds    : \n";

    cin >> numOfSeconds;

    float result = totalSeconds(numOfDays, numOfHours, numOfMinutes, numOfSeconds);

    cout << "Total Seconds : " << result;

    return 0;

}

#43

void totalSecondsCalc(float totalseconds)

{

    const float secondsPerDay = 24 \* 60 \* 60;

    const float secondsPerHour = 60 \* 60;

    const float secondsPerMinute = 60;

    float reminder;

    float numOfSeconds;

    float numOfDays = floor(totalseconds / secondsPerDay);

    reminder = fmod(totalseconds, secondsPerDay);

    float numOfHour = floor(reminder / secondsPerHour);

    reminder = fmod(reminder, secondsPerHour);

    float numOfMinutes = floor(reminder / secondsPerMinute);

    reminder = fmod(reminder, secondsPerMinute);

    numOfSeconds = reminder;

    cout << numOfDays << " : " << numOfHour << ": " << numOfMinutes << ": " << numOfSeconds;

}

int main()

{

    float totalSeconds;

    cout << "Please enter totalSeconds   : \n ";

    cin >> totalSeconds;

    totalSecondsCalc(totalSeconds);

    return 0;

}

Conditional Statements :If, else statements

#4

struct strInfo

{

    short Age;

    bool IsDriverLicenseHave;

};

void chceckIfisHired(strInfo info)

{

    if (info.Age > 21 && info.IsDriverLicenseHave)

    {

        cout << "Hire";

    }

    else

    {

        cout << "Rejected";

    }

}

void readInfo(strInfo &info)

{

    cout << "Enter Your age  ?" << "\n";

    cin >> info.Age;

    cout << "Do you have Driver License  ,1 (yes) or 0 (no)." << "\n";

    cin >> info.IsDriverLicenseHave;

}

int main()

{

    strInfo drivingInfo;

    readInfo(drivingInfo);

    chceckIfisHired(drivingInfo);

}

#8

void printStudentScore(float avg)

{

    if (avg >= 50)

    {

        cout << "PASS";

    }

    else

    {

        cout << "FAIL";

    }

}

int main()

{

    int mark;

    cout << "Enter Your mark ? " << "\n";

    cin >> mark;

    printStudentScore(mark);

}

#11

void readMarks(int marks[3])

{

    cout << "Enter Mark1 ?" << "\n";

    cin >> marks[0];

    cout << "Enter Mark2  ? " << "\n";

    cin >> marks[1];

    cout << "Enter Mark3  ? " << "\n";

    cin >> marks[2];

}

float calculateAvgOfMarks(int marks[3])

{

    return (marks[0] + marks[1] + marks[2]) / 3;

}

void printStudentScore(float avg)

{

    if (avg >= 50)

    {

        cout << "PASS";

    }

    else

    {

        cout << "FAIL";

    }

}

int main()

{

    int marks[3];

    readMarks(marks);

    float avg = calculateAvgOfMarks(marks);

    printStudentScore(avg);

}

#24

void validateAge(short age)

{

    if (age >= 18 && age <= 48)

    {

        cout << "Valid Age";

    }

    else

    {

        cout << "Invalid Age";

    }

}

int main()

{

    short age;

    cout << "Enter Your Age ? " << "\n";

    cin >> age;

    validateAge(age);

}

#49

string validatePIN(string pinCode)

{

    string balance = "7500";

    if (pinCode == "1234")

    {

        return "Your Balance is : " + balance;

    }

    else

    {

        return "Wrong PIN";

    }

}

int main()

{

    string pin;

    cout << "enter PIN Code ?" << "\n";

    getline(cin, pin);

    cout << validatePIN(pin) << endl;

}

#33

void printGrade(short grade)

{

    if (grade >= 90)

    {

        cout << "A" << "\n";

    }

    else if (grade >= 80)

    {

        cout << "B" << "\n";

    }

    else if (grade >= 70)

    {

        cout << "C" << "\n";

    }

    else if (grade >= 60)

    {

        cout << "D" << "\n";

    }

    else if (grade >= 50)

    {

        cout << "E" << "\n";

    }

    else

    {

        cout << "F" << "\n";

    }

}

int main()

{

    short grade;

    cout << "Enter Your grade of 100 ? " << "\n";

    cin >> grade;

    printGrade(grade);

}

#34

float getPercentage(float totalSales)

{

    float percentage;

    if (totalSales >= 1e6)

    {

        percentage = 0.01;

    }

    else if (totalSales >= 5e5)

    {

        percentage = 0.02;

    }

    else if (totalSales >= 1e5)

    {

        percentage = 0.03;

    }

    else if (totalSales >= 1e4)

    {

        percentage = 0.05;

    }

    else

        percentage = 0;

    return percentage;

}

int main()

{

    float totalSales;

    cout << "Enter Your  Total Sales ?  " << "\n";

    cin >> totalSales;

    float percent = getPercentage(totalSales);

    float totalCommision = totalSales \* percent;

    cout << totalCommision << endl;

}

#36

struct strCalculation

{

    float Number1;

    float Number2;

    string Operation;

};

void readCalculationInfo(strCalculation &calculation)

{

    cout << "Enter N1 ? " << "\n";

    cin >> calculation.Number1;

    cout << "Enter N2 ? " << "\n";

    cin >> calculation.Number2;

    cout << "Enter Operation  to apply into two numbers ? " << "\n";

    cin >> calculation.Operation;

}

float getResultFromOperation(strCalculation calc)

{

    if (calc.Operation == "\*")

        return calc.Number1 \* calc.Number2;

    else if (calc.Operation == "-")

        return calc.Number1 - calc.Number2;

    else if (calc.Operation == "+")

        return calc.Number1 + calc.Number2;

    else if (calc.Operation == "/")

        return calc.Number1 / calc.Number2;

    else

        return 1;

}

int main()

{

    strCalculation calculation;

    readCalculationInfo(calculation);

    cout << calculation.Number1 << calculation.Operation << calculation.Number2 << "=" << getResultFromOperation(calculation);

}

#44

void printDay(short day)

{

    if (day == 1)

    {

        cout << "Sunday";

    }

    else if (day == 2)

    {

        cout << "Monday";

    }

    else if (day == 3)

    {

        cout << "Tuesday";

    }

    else if (day == 4)

    {

        cout << "Wednesday";

    }

    else if (day == 5)

    {

        cout << "Thursday";

    }

    else if (day == 6)

    {

        cout << "Friday";

    }

    else if (day == 7)

    {

        cout << "Saturday";

    }

}

int main()

{

    short day;

    cout << "Enter day in number ? " << "\n";

    cin >> day;

    printDay(day);

}

#45

void printMonth(short month)

{

    if (month == 1)

    {

        cout << "January";

    }

    else if (month == 2)

    {

        cout << "February";

    }

    else if (month == 3)

    {

        cout << "March";

    }

    else if (month == 4)

    {

        cout << "April";

    }

    else if (month == 5)

    {

        cout << "May";

    }

    else if (month == 6)

    {

        cout << "Jun";

    }

    else if (month == 7)

    {

        cout << "July";

    }

    else if (month == 8)

    {

        cout << "August";

    }

    else if (month == 9)

    {

        cout << "Septmeper";

    }

    else if (month == 10)

    {

        cout << "October";

    }

    else if (month == 11)

    {

        cout << "November";

    }

    else if (month == 12)

    {

        cout << "December";

    }

    else

    {

        cout << "Wrong Month ";

    }

}

int main()

{

    short month;

    cout << "Enter  month ? " << "\n";

    cin >> month;

    printMonth(month);

}

Switch

#36

struct strCalculation

{

    float Number1;

    float Number2;

    char Operation;

};

void readCalculationInfo(strCalculation &calculation)

{

    cout << "Enter N1 ? " << "\n";

    cin >> calculation.Number1;

    cout << "Enter N2 ? " << "\n";

    cin >> calculation.Number2;

    cout << "Enter Operation  to apply into two numbers ? " << "\n";

    cin >> calculation.Operation;

}

float getResultFromOperation(strCalculation calc)

{

    switch (calc.Operation)

    {

    case '\*':

        return calc.Number1 \* calc.Number2;

        break;

    case '-':

        return calc.Number1 - calc.Number2;

        break;

    case '+':

        return calc.Number1 + calc.Number2;

        break;

    case '/':

        return calc.Number1 / calc.Number2;

        break;

    default:

        break;

    }

}

int main()

{

    strCalculation calculation;

    readCalculationInfo(calculation);

    cout << calculation.Number1 << calculation.Operation << calculation.Number2 << "=" << getResultFromOperation(calculation);

}

#44

void printDay(short day)

{

    switch (day)

    {

    case 1:

        cout << "Sunday";

        break;

    case 2:

        cout << "Monday";

        break;

    case 3:

        cout << "Tuesday";

        break;

    case 4:

        cout << "Wednesday";

        break;

    case 5:

        cout << "Thursday";

        break;

    case 6:

        cout << "Friday";

        break;

    case 7:

        cout << "Saturday";

        break;

    }

}

int main()

{

    short day;

    cout << "Enter day in number ? " << "\n";

    cin >> day;

    printDay(day);

}

#45

void printMonth(short month)

{

    switch (month)

    {

    case 1:

        cout << "January";

        break;

    case 2:

        cout << "February";

        break;

    case 3:

        cout << "March";

        break;

    case 4:

        cout << "April";

        break;

    case 5:

        cout << "May";

        break;

    case 6:

        cout << "Jun";

        break;

    case 7:

        cout << "July";

        break;

    case 8:

        cout << "August";

        break;

    case 9:

        cout << "Septmeper";

        break;

    case 10:

        cout << "October";

        break;

    case 11:

        cout << "November";

        break;

    case 12:

        cout << "December";

        break;

    default:

        cout << "Wrong Month ";

        break;

    }

}

int main()

{

    short month;

    cout << "Enter  month ? " << "\n";

    cin >> month;

    printMonth(month);

}

For Loops

#26

int ReadNumber()

{

    int number;

    cout << "enter Scope of number to perint" << endl;

    cin >> number;

    return number;

}

void printNumbers1\_N(int N)

{

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    for (int i = 1; i <= N; i++)

    {

        cout << i << "\n";

    }

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

int main()

{

    printNumbers1\_N(ReadNumber());

}

#27

int ReadNumber()

{

    int number;

    cout << "enter Scope of number to perint" << endl;

    cin >> number;

    return number;

}

void printReversedNumbersN\_1(int N)

{

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    for (int i = N; i >= 1; i--)

    {

        cout << i << "\n";

    }

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

int main()

{

    printReversedNumbersN\_1(ReadNumber());

}

#28

int ReadNumber()

{

    int number;

    cout << "enter Scope of number to print" << endl;

    cin >> number;

    return number;

}

int sumOddNumbers(int N)

{

    int sum = 0;

    for (int i = 1; i <= N; i = i + 2)

    {

        sum += i;

    }

    return sum;

}

int main()

{

    int selectedNumber = ReadNumber();

    int num = sumOddNumbers(selectedNumber);

    cout << "Sum of Odd of " << selectedNumber << "Is : " << num;

}

#29

int ReadNumber()

{

    int number;

    cout << "enter Scope of number to print" << endl;

    cin >> number;

    return number;

}

int sumEvenNumbers(int N)

{

    int sum = 0;

    for (int i = 0; i <= N; i = i + 2)

    {

        sum += i;

    }

    return sum;

}

int main()

{

    int selectedNumber = ReadNumber();

    int num = sumEvenNumbers(selectedNumber);

    cout << "Sum of Even of " << selectedNumber << "Is : " << num;

}

#30

int ReadNumber()

{

    int number;

    cout << "enter Scope of number to print" << endl;

    cin >> number;

    return number;

}

int FactorialOfN(int N)

{

    double factorial = 1;

    for (int i = N; i >= 1; i--)

    {

        factorial = factorial \* i;

    }

    return factorial;

}

int main()

{

    int selectedNumber = ReadNumber();

    int fact = FactorialOfN(selectedNumber);

    cout << "Factorial of  " << selectedNumber << "Is : " << fact;

}

#32

struct strPowerInputs

{

    int N;

    int M;

};

strPowerInputs ReadNumber()

{

    strPowerInputs pm;

    cout << "enter  number to print" << endl;

    cin >> pm.N;

    cout << "enter power M" << endl;

    cin >> pm.M;

    return pm;

}

int PowerOfM(strPowerInputs pm)

{

    int powerRes = 1;

    for (int i = 1; i <= pm.M; i++)

    {

        powerRes = powerRes \* pm.N;

    }

    return powerRes;

}

int main()

{

    strPowerInputs selectedNumber = ReadNumber();

    int powerres = PowerOfM(selectedNumber);

    cout << selectedNumber.N << "power " << selectedNumber.M << "Is : " << powerres;

}

#46

void printA\_Z()

{

    for (int i = 65; i <= 90; i++)

    {

        cout << char(i) << "\n";

    }

}

int main()

{

    printA\_Z();

}

While Loop

#26

int ReadNumber()

{

    int number;

    cout << "enter Scope of number to perint" << endl;

    cin >> number;

    return number;

}

void printNumbers1\_N(int N)

{

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    int i = 1;

    while (i <= N)

    {

        cout << i << "\n";

        i++;

    }

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

int main()

{

    printNumbers1\_N(ReadNumber());

}

#27

int ReadNumber()

{

    int number;

    cout << "enter Scope of number to perint" << endl;

    cin >> number;

    return number;

}

void printReversedNumbersN\_1(int N)

{

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    int i = N;

    while (i >= 1)

    {

        cout << i << "\n";

        i--;

    }

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

int main()

{

    printReversedNumbersN\_1(ReadNumber());

}

#28

int ReadNumber()

{

    int number;

    cout << "enter Scope of number to print" << endl;

    cin >> number;

    return number;

}

int sumOddNumbers(int N)

{

    int sum = 0;

    int i = 1;

    while (i <= N)

    {

        sum += i;

        i = i + 2;

    }

    return sum;

}

int main()

{

    int selectedNumber = ReadNumber();

    int num = sumOddNumbers(selectedNumber);

    cout << "Sum of Odd of " << selectedNumber << "Is : " << num;

}

#29

int ReadNumber()

{

    int number;

    cout << "enter Scope of number to print" << endl;

    cin >> number;

    return number;

}

int sumEvenNumbers(int N)

{

    int sum = 0;

    int i = 0;

    while (i <= N)

    {

        sum += i;

        i = i + 2;

    }

    return sum;

}

int main()

{

    int selectedNumber = ReadNumber();

    int num = sumEvenNumbers(selectedNumber);

    cout << "Sum of Even of " << selectedNumber << "Is : " << num;

}

#30

int ReadNumber()

{

    int number;

    cout << "enter Scope of number to print" << endl;

    cin >> number;

    return number;

}

int FactorialOfN(int &N)

{

    double factorial = 1;

    while (N < 0)

    {

        cout << "InValid Number please enter positive number \n";

        N = ReadNumber();

    }

    for (int i = N; i >= 1; i--)

    {

        factorial = factorial \* i;

    }

    return factorial;

}

int main()

{

    int selectedNumber = ReadNumber();

    int fact = FactorialOfN(selectedNumber);

    cout << "Factorial of  " << selectedNumber << " Is : " << fact;

}

#32

struct strPowerInputs

{

    int N;

    int M;

};

strPowerInputs ReadNumber()

{

    strPowerInputs pm;

    cout << "enter  number to print" << endl;

    cin >> pm.N;

    cout << "enter power M" << endl;

    cin >> pm.M;

    return pm;

}

int PowerOfM(strPowerInputs pm)

{

    int powerRes = 1;

    int i = 1;

    while (i <= pm.M)

    {

        powerRes = powerRes \* pm.N;

        i++;

    }

    return powerRes;

}

int main()

{

    strPowerInputs selectedNumber = ReadNumber();

    int powerres = PowerOfM(selectedNumber);

    cout << selectedNumber.N << "power " << selectedNumber.M << "Is : " << powerres;

}

#37

int readNumber()

{

    int num;

    cout << "Enter Any Number, -99 will quit the program " << endl;

    cin >> num;

    return num;

}

int calculateSumUntil(int &N)

{

    int sum = 0;

    while (N != -99)

    {

        /\* code \*/

        sum += N;

        N = readNumber();

    }

    return sum;

}

int main()

{

    int num = readNumber();

    int res = calculateSumUntil(num);

    cout << "Sum is : " << res << endl;

}

#46

void printA\_Z()

{

    int i = 65;

    while (i <= 90)

    {

        /\* code \*/

        cout << char(i) << "\n";

        i++;

    }

}

int main()

{

    printA\_Z();

}

#50

tring readPinCode()

{

    string pin;

    cout << "enter PIN Code , 1234 get balance?" << "\n";

    getline(cin, pin);

    return pin;

}

string validatePIN(string &pinCode)

{

    string balance = "7500";

    while (pinCode != "1234")

    {

        /\* code \*/

        pinCode = readPinCode();

    }

    return "Your Balance is : " + balance;

}

int main()

{

    string pin = readPinCode();

    cout << validatePIN(pin) << endl;

}

\_

Course 4 :Problems & Algorithms :الحلول المثالية

#4 + #5

struct strInfo

{

    short Age;

    bool HasDriverLicense;

    bool HasRecommendation;

};

strInfo readInfo()

{

    strInfo info;

    cout << "Enter Your age  ?" << "\n";

    cin >> info.Age;

    cout << "Do you have Driver License  ,1 (yes) or 0 (no)." << "\n";

    cin >> info.HasDriverLicense;

    cout << "Do you have  Recommendation" << "\n";

    cin >> info.HasRecommendation;

    return info;

}

bool chceckIfisHired(strInfo info)

{

    if (info.HasRecommendation)

    {

        return true;

    }

    else

    {

        return (info.Age > 21 && info.HasDriverLicense);

    }

}

void printResult(strInfo info)

{

    bool isAccepted = chceckIfisHired(info);

    if (isAccepted)

    {

        cout << "\nHire\n";

    }

    else

    {

        cout << "\nRejected\n";

    }

}

int main()

{

    strInfo info = readInfo();

    printResult(info);

}

#6

enum enPrinDir

{

    FirstNameLastName,

    LastNameFistName

};

struct strInfo

{

    string FirstName;

    string LastName;

};

strInfo readInfo()

{

    strInfo info;

    cout << "Enter Your  First Name  ?" << "\n";

    cin >> info.FirstName;

    cout << "Enter Your Last Name " << "\n";

    cin >> info.LastName;

    return info;

}

string getFullName(strInfo info, enPrinDir printDir)

{

    string fullName = "";

    if (printDir == enPrinDir::LastNameFistName)

        fullName = info.LastName + " " + info.FirstName;

    else

        fullName = info.FirstName + " " + info.LastName;

    return fullName;

}

void printFullName(string fullname)

{

    cout << "Your Full Name is: " << fullname << endl;

}

int main()

{

    strInfo info = readInfo();

    string fullname = getFullName(info, enPrinDir::FirstNameLastName);

    printFullName((fullname));

}

#7

int readNumber()

{

    int num;

    cout << "Enter  A Number ?" << "\n";

    cin >> num;

    return num;

}

float calculateHalfNumber(int num)

{

    float halfNum = (float)num / 2;

    return halfNum;

}

void printResults(int num)

{

    cout << "Half of " << num << " is : " << calculateHalfNumber(num) << endl;

}

int main()

{

    int num = readNumber();

    printResults(num);

}

#8

using namespace std;

enum enPassFail

{

    Pass = 1,

    Fail = 2

};

int readMark()

{

    int mark;

    cout << "Enter  Your Mark from 100 ?" << "\n";

    cin >> mark;

    return mark;

}

enPassFail checkMark(int mark)

{

    if (mark >= 50)

        return enPassFail::Pass;

    else

        return enPassFail::Fail;

}

void printResults(int mark)

{

    if (checkMark(mark) == enPassFail::Pass)

        cout << "\n You Passed" << endl;

    else

        cout << "\n You Failed" << endl;

}

int main()

{

    int mark = readMark();

    printResults(mark);

}

#9

void readNumbers(int &num1, int &num2, int &num3)

{

    cout << "Please Enter Num 1 : \n";

    cin >> num1;

    cout << "Please Enter Num 2 : \n";

    cin >> num2;

    cout << "Please Enter Num 3 : \n";

    cin >> num3;

}

int sumOf3Numbers(int &num1, int &num2, int &num3)

{

    int sum = num1 + num2 + num3;

    return sum;

}

void printResults(int totalSum)

{

    string result = "Total Sum of numbers is : ";

    cout << result << totalSum;

}

int main()

{

    int num1, num2, num3;

    readNumbers(num1, num2, num3);

    printResults(sumOf3Numbers(num1, num2, num3));

}

#10

using namespace std;

void readNumbers(int &num1, int &num2, int &num3)

{

    cout << "Please Enter Num 1 : \n";

    cin >> num1;

    cout << "Please Enter Num 2 : \n";

    cin >> num2;

    cout << "Please Enter Num 3 : \n";

    cin >> num3;

}

int sumOf3Numbers(int &num1, int &num2, int &num3)

{

    int sum = num1 + num2 + num3;

    return sum;

}

float calculateAvg(int &num1, int &num2, int &num3)

{

    float Avg = (float)sumOf3Numbers(num1, num2, num3) / 3;

    return Avg;

}

void printResults(float Average)

{

    string result = "Average of numbers is : ";

    cout << result << Average;

}

int main()

{

    int num1, num2, num3;

    readNumbers(num1, num2, num3);

    printResults(calculateAvg(num1, num2, num3));

}

#11

enum enPassFail

{

    Pass = 1,

    Fail = 2

};

void readNumbers(int &num1, int &num2, int &num3)

{

    cout << "Please Enter Num 1 : \n";

    cin >> num1;

    cout << "Please Enter Num 2 : \n";

    cin >> num2;

    cout << "Please Enter Num 3 : \n";

    cin >> num3;

}

int sumOf3Numbers(int &num1, int &num2, int &num3)

{

    int sum = num1 + num2 + num3;

    return sum;

}

float calculateAvg(int &num1, int &num2, int &num3)

{

    float Avg = (float)sumOf3Numbers(num1, num2, num3) / 3;

    return Avg;

}

enPassFail checkAverage(float average)

{

    if (average >= 50)

        return enPassFail::Pass;

    else

        return enPassFail::Fail;

}

void printResults(float average)

{

    cout << "\n Your Average is : " << average << endl;

    if (checkAverage(average) == enPassFail::Pass)

        cout << "\n You Passed " << endl;

    else

        cout << "\n You Failed " << endl;

}

int main()

{

    int num1, num2, num3;

    readNumbers(num1, num2, num3);

    printResults(calculateAvg(num1, num2, num3));

}

#12

void readeNumbers(int &num1, int &num2)

{

    cout << "Enter num1 : " << endl;

    cin >> num1;

    cout << "Enter num2 : " << endl;

    cin >> num2;

}

int maxOf2Numbers(int num1, int num2)

{

    int max;

    if (num1 > num2) // num1 is greater

        max = num1;

    else if (num1 < num2) // num2 is greater

        max = num2;

    else // equality so take any one

        max = num1;

    return max;

}

void printResults(int max)

{

    cout << "\n Max Number is " << max;

}

int main()

{

    int num1, num2;

    readeNumbers(num1, num2);

    printResults(maxOf2Numbers(num1, num2));

}

#13

using namespace std;

void readeNumbers(int &num1, int &num2, int &num3)

{

    cout << "Enter num1 : " << endl;

    cin >> num1;

    cout << "Enter num2 : " << endl;

    cin >> num2;

    cout << "Enter num3 : " << endl;

    cin >> num3;

}

int maxOf2Numbers(int num1, int num2)

{

    int max;

    if (num1 > num2) // num1 is greater

        max = num1;

    else if (num1 < num2) // num2 is greater

        max = num2;

    else // equality so take any one

        max = num1;

    return max;

}

int maxOf3Numbers(int num1, int num2, int num3)

{

    int max;

    max = maxOf2Numbers(num1, num2);

    max = maxOf2Numbers(max, num3);

    return max;

}

void printResults(int max)

{

    cout << "\n Max Number is " << max;

}

int main()

{

    int num1, num2, num3;

    readeNumbers(num1, num2, num3);

    printResults(maxOf3Numbers(num1, num2, num3));

}

#14

void readeNumbers(int &num1, int &num2)

{

    cout << "Enter num1 : " << endl;

    cin >> num1;

    cout << "Enter num2 : " << endl;

    cin >> num2;

}

void swapNumbers(int &num1, int &num2)

{

    int temp = num1;

    num1 = num2;

    num2 = temp;

}

void printNumbers(int num1, int num2)

{

    cout << "\n num1 is  " << num1 << endl;

    cout << "\n num2 is  " << num2 << endl;

}

int main()

{

    int num1, num2;

    readeNumbers(num1, num2);

    cout << "\n Before Swapping " << endl;

    printNumbers(num1, num2);

    cout << "\n After Swapping " << endl;

    swapNumbers(num1, num2);

    printNumbers(num1, num2);

}

#15

struct strRectangle

{

    /\* data \*/

    float Width;

    float Length;

};

strRectangle ReadInputs()

{

    strRectangle rectangleInfo;

    cout << "Please enter rectangle  width ? \n";

    cin >> rectangleInfo.Width;

    cout << "Please enter  rectangle length? \n";

    cin >> rectangleInfo.Length;

    return rectangleInfo;

}

float calculateRectangleArea(strRectangle rectangleInfo)

{

    float rectangleArea = rectangleInfo.Width \* rectangleInfo.Length;

    return rectangleArea;

}

void printResult(float area)

{

    cout << "\n Rectangle area is :" << area;

}

int main()

{

    strRectangle strInfo = ReadInputs();

    printResult(calculateRectangleArea(strInfo));

    return 0;

}

#16

struct strRectangle

{

    /\* data \*/

    float Width;

    float Length;

    float rectangleDiagonal;

    float rectangleSide;

};

strRectangle ReadInputs()

{

    strRectangle rectangleInfo;

    cout << "Please enter rectangle  diagonal ? \n";

    cin >> rectangleInfo.rectangleDiagonal;

    cout << "Please enter  rectangle side? \n";

    cin >> rectangleInfo.rectangleSide;

    return rectangleInfo;

}

float calculateRectangleArea(strRectangle rectangleInfo)

{

    float rectangleArea =   rectangleInfo.rectangleSide \* sqrt(pow(rectangleInfo.rectangleDiagonal, 2) - pow(rectangleInfo.rectangleSide, 2));

    return rectangleArea;

}

void printResult(float area)

{

    cout << "\n Rectangle area is :" << area;

}

int main()

{

    strRectangle strInfo = ReadInputs();

    printResult(calculateRectangleArea(strInfo));

    return 0;

}

#17

struct strTriangle

{

    /\* data \*/

    float base;

    float height;

};

strTriangle ReadInputs()

{

    strTriangle triangleInfo;

    cout << "Please enter triangle  base ? \n";

    cin >> triangleInfo.base;

    cout << "Please enter  triangle height? \n";

    cin >> triangleInfo.height;

    return triangleInfo;

}

float calculateTriangleArea(strTriangle triangleInfo)

{

    float triangleArea = 0.5f \* triangleInfo.base \* triangleInfo.height; // f==>float

    return triangleArea;

}

void printResult(float area)

{

    cout << "\n Triangle area is :" << area;

}

int main()

{

    strTriangle strInfo = ReadInputs();

    printResult(calculateTriangleArea(strInfo));

    return 0;

}

#18

void ReadInputs(float &radius)

{

    cout << "Please enter radius   : \n ";

    cin >> radius;

}

float CalculateCircleArea(float radius)

{

    const float PI = 3.14;

    float area = PI \* pow(radius, 2);

    return area;

}

void PrintResult(float circleArea)

{

    cout << "Area of circle  : " << circleArea << endl;

}

int main()

{

    float circleRadius;

    ReadInputs(circleRadius);

    PrintResult(CalculateCircleArea(circleRadius));

    return 0;

}

#19

float ReadDiameter()

{

    float diameter;

    cout << "Please enter diameter   : \n ";

    cin >> diameter;

    return diameter;

}

float CalculateCircleAreaByDiameter(float diameter)

{

    const float PI = 3.14;

    float radius = diameter / 2;

    float area = PI \* pow(radius, 2);

    return area;

}

void PrintResult(float circleArea)

{

    cout << "Area of circle  : " << circleArea << endl;

}

int main()

{

    float diameter = ReadDiameter();

    PrintResult(CalculateCircleAreaByDiameter(diameter));

    return 0;

}

#20

float ReadSquareSide()

{

    float squareSide;

    cout << "Please enter squareSide   : \n ";

    cin >> squareSide;

    return squareSide;

}

float CalculateCircleinSquare(float squareSide)

{

    const float PI = 3.14;

    float circleArea = (PI \* squareSide \* squareSide) / 4;

    return circleArea;

}

void PrintResult(float circleArea)

{

    cout << "Area of circle  : " << circleArea << endl;

}

int main()

{

    float squareSide = ReadSquareSide();

    PrintResult(CalculateCircleinSquare(squareSide));

    return 0;

}

#21

using namespace std;

float ReadCircumeference()

{

    float circumeference;

    cout << "Please enter circumeference   : \n ";

    cin >> circumeference;

    return circumeference;

}

float circleAreaByCircumeference(float circleCircumference)

{

    const float PI = 3.14;

    float circleArea = (circleCircumference \* circleCircumference) / (4 \* PI);

    return circleArea;

}

void PrintResult(float circleArea)

{

    cout << "Area of circle  : " << circleArea << endl;

}

int main()

{

    float circleCircumference = ReadCircumeference();

    PrintResult(circleAreaByCircumeference(circleCircumference));

    return 0;

}

#22

struct strIsoScelesTriangle

{

    float triangleequalSides;

    float trianglebase;

};

strIsoScelesTriangle ReadInputs()

{

    strIsoScelesTriangle IsoScelesTriangle;

    cout << "Please enter triangle sides length ,1-triangleequalSides 2-  trianglebase? \n";

    cin >> IsoScelesTriangle.triangleequalSides;

    cin >> IsoScelesTriangle.trianglebase;

    return IsoScelesTriangle;

}

float circleArea(strIsoScelesTriangle IsoScelesTriangle)

{

    const float PI = 3.14;

    float circleArea = (PI \* IsoScelesTriangle.trianglebase \* IsoScelesTriangle.trianglebase / 4) \* ((2 \* IsoScelesTriangle.triangleequalSides - IsoScelesTriangle.trianglebase) / (2 \* IsoScelesTriangle.triangleequalSides + IsoScelesTriangle.trianglebase));

    return circleArea;

}

void PrintResult(float circleArea)

{

    cout << "Area of circle  : " << circleArea << endl;

}

int main()

{

    strIsoScelesTriangle IsoScelesTriangle = ReadInputs();

    PrintResult(circleArea(IsoScelesTriangle));

    return 0;

}

#23

struct strArbitraryTriangle

{

    float Side1;

    float Side2;

    float Side3;

};

strArbitraryTriangle ReadInputs()

{

    strArbitraryTriangle ArbitraryTriangle;

    cout << "Please enter triangleSides A,B,C  in order   : \n ";

    cin >> ArbitraryTriangle.Side1;

    cin >> ArbitraryTriangle.Side2;

    cin >> ArbitraryTriangle.Side3;

    return ArbitraryTriangle;

}

float circleAreaByArbitraryTriangle(strArbitraryTriangle ArbitraryTriangle)

{

    const float PI = 3.14;

    double HalfOftriangleCircumference = (ArbitraryTriangle.Side1 + ArbitraryTriangle.Side2 + ArbitraryTriangle.Side3) / 2;

    double radius = ArbitraryTriangle.Side1 \* ArbitraryTriangle.Side2 \* ArbitraryTriangle.Side3 / (4 \* sqrt(HalfOftriangleCircumference \* (HalfOftriangleCircumference - ArbitraryTriangle.Side1) \* (HalfOftriangleCircumference - ArbitraryTriangle.Side2) \* (HalfOftriangleCircumference - ArbitraryTriangle.Side3)));

    double area = PI \* pow(radius, 2);

    return area;

}

void PrintResult(float circleArea)

{

    cout << "Area of circle  : " << circleArea << endl;

}

int main()

{

    strArbitraryTriangle arbitraryTriangle = ReadInputs();

    PrintResult(circleAreaByArbitraryTriangle(arbitraryTriangle));

}

#24

short ReadAge()

{

    short age;

    cout << "Enter Your Age ? " << "\n";

    cin >> age;

    return age;

}

bool validateNumberInRange(short number, short from, short to)

{

    return (number >= from && number <= to);

}

void PrintResult(short age)

{

    if (validateNumberInRange(age, 18, 45))

        cout << age << " is valid age" << endl;

    else

        cout << age << " is invalid age" << endl;

}

int main()

{

    short age = ReadAge();

    PrintResult(age);

}

#25

short ReadAge()

{

    short age;

    cout << "Enter Your Age between 18 and 45 ? " << "\n";

    cin >> age;

    return age;

}

bool validateNumberInRange(short number, short from, short to)

{

    return (number >= from && number <= to);

}

short ReadUntiAgeBetween(short from, short to)

{

    short age = 0;

    do

    {

        age = ReadAge();

    } while (!validateNumberInRange(age, from, to));

    return age;

}

void PrintResult(short age)

{

    cout << "\n your age is : " << age << endl;

}

int main()

{

    PrintResult(ReadUntiAgeBetween(18, 45));

}

#26

int ReadNumber()

{

    int number;

    cout << "enter Scope of number to perint" << endl;

    cin >> number;

    return number;

}

void printRangeFrom1toN\_UsingWhile(int N)

{

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    int i = 1;

    while (i <= N)

    {

        cout << i << "\n";

        i++;

    }

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

void printRangeFrom1toN\_UsingDoWhile(int N)

{

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    int i = 1;

    do

    {

        cout << i << "\n";

        i++;

    }

    while (i <= N);

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

void printRangeFrom1toN\_UsingFor(int N)

{

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    for (int i = 1; i <= N; i++)

    {

        cout << i << "\n";

    }

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

int main()

{

    int n = ReadNumber();

    printRangeFrom1toN\_UsingWhile(n);

    printRangeFrom1toN\_UsingDoWhile(n);

    printRangeFrom1toN\_UsingFor(n);

}

#27

int ReadNumber()

{

    int number;

    cout << "enter Scope of number to perint" << endl;

    cin >> number;

    return number;

}

void printReversedNumbersNto1\_UsingFor(int N)

{

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    for (int i = N; i >= 1; i--)

    {

        cout << i << "\n";

    }

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

void printReversedNumbersNto1\_UsingWhile(int N)

{

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    int i = N;

    while (i >= 1)

    {

        cout << i << "\n";

        i--;

    }

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

void printReversedNumbersNto1\_UsingDoWhile(int N)

{

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    int i = N;

    do

    {

        cout << i << "\n";

        i--;

    } while (i >= 1);

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

int main()

{

    int n = ReadNumber();

    printReversedNumbersNto1\_UsingFor(n);

    printReversedNumbersNto1\_UsingWhile(n);

    printReversedNumbersNto1\_UsingDoWhile(n);

}

#28

enum enOddOrEven

{

    Odd = 1,

    Even = 2

};

int ReadNumber()

{

    int number;

    cout << "enter Scope of number to print" << endl;

    cin >> number;

    return number;

}

enOddOrEven CheckOddOrEven(int number)

{

    if (number % 2 == 0)

        return enOddOrEven::Even;

    else

        return enOddOrEven ::Odd;

}

int sumOddNumbersFrom1toN\_UsingFor(int N)

{

    int sum = 0;

    cout << "\n Sum odd numbers using ForLoop is \n";

    for (int i = 1; i <= N; i++)

    {

        if (CheckOddOrEven(i) == enOddOrEven::Odd)

        {

            sum += i;

        }

    }

    return sum;

}

int sumOddNumbersFrom1toN\_UsingWhile(int N)

{

    cout << "\n Sum odd numbers using While Loop is \n";

    int sum = 0;

    int i = 1;

    while (i <= N)

    {

        if (CheckOddOrEven(i) == enOddOrEven::Odd)

        {

            sum += i;

        }

        i++;

    }

    return sum;

}

int sumOddNumbersFrom1toN\_UsingDoWhile(int N)

{

    int sum = 0;

    int i = 1;

    cout << "\n Sum odd numbers using do While Loop is \n";

    do

    {

        if (CheckOddOrEven(i) == enOddOrEven::Odd)

        {

            sum += i;

        }

        i++;

    } while (i <= N);

    return sum;

}

int main()

{

    int selectedNumber = ReadNumber();

    cout << sumOddNumbersFrom1toN\_UsingFor(selectedNumber) << endl;

    cout << sumOddNumbersFrom1toN\_UsingDoWhile(selectedNumber) << endl;

    cout << sumOddNumbersFrom1toN\_UsingWhile(selectedNumber) << endl;

}

#31

int ReadNumber()

{

    int number = 0;

    do

    {

        cout << "enter  number to print" << endl;

        cin >> number;

    } while (number <= 0);

    return number;

}

int calculatePowerOfN(int N, int power)

{

    int nPower = 1;

    for (int i = 1; i <= power; i++)

    {

        nPower \*= N;

    }

    return nPower;

}

void PowerOf2\_3\_4(int number)

{

    cout << "power 2: " << calculatePowerOfN(number, 2) << "\n";

    cout << "power 3: " << calculatePowerOfN(number, 3) << "\n";

    cout << "power 4: " << calculatePowerOfN(number, 4) << "\n";

}

int main()

{

    PowerOf2\_3\_4(ReadNumber());

    return 0;

}

#32

int ReadNumber()

{

    int N = 0;

    do

    {

        cout << "enter  number to print " << endl;

        cin >> N;

    } while (N < 0);

    return N;

}

int ReadPower()

{

    int power = 0;

    do

    {

        cout << "enter  power to print " << endl;

        cin >> power;

    } while (power < 0);

    return power;

}

int calculatePowerOfN(int N, int power)

{

    int nPower = 1;

    if (power == 0)

    {

        return 1;

    }

    for (int i = 1; i <= power; i++)

    {

        nPower \*= N;

    }

    return nPower;

}

int main()

{

    int N = ReadNumber();

    int P = ReadPower();

    cout << endl

         << "result is " << calculatePowerOfN(N, P) << endl;

    return 0;

}

#33

short ReadNumberInRange(short from, short to)

{

    short grade;

    do

    {

        cout << "Enter Your grade of 100 ? " << "\n";

        cin >> grade;

    } while (grade < from || grade > to);

    return grade;

}

// validation

char GetGradeLetter(short grade)

{

    if (grade >= 90)

        return 'A';

    else if (grade >= 80)

        return 'B';

    else if (grade >= 70)

        return 'C';

    else if (grade >= 60)

        return 'D';

    else if (grade >= 50)

        return 'E';

    else

        return 'F';

}

int main()

{

    cout << GetGradeLetter(ReadNumberInRange(0, 100)) << endl;

}

short ReadNumberInRange(short from, short to)

{

    short grade;

    do

    {

        cout << "Enter Your grade of 100 ? " << "\n";

        cin >> grade;

    } while (grade < from || grade > to);

    return grade;

}

// validation

char GetGradeLetter(short grade)

{

    if (grade >= 90)

        return 'A';

    else if (grade >= 80)

        return 'B';

    else if (grade >= 70)

        return 'C';

    else if (grade >= 60)

        return 'D';

    else if (grade >= 50)

        return 'E';

    else

        return 'F';

}

int main()

{

    cout << GetGradeLetter(ReadNumberInRange(0, 100)) << endl;

}

#34

float ReadTotalSales()

{

    float totalSales;

    cout << "Enter Your  Total Sales ?  " << "\n";

    cin >> totalSales;

    return totalSales;

}

float GetCommissionPercentage(float totalSales)

{

    float percentage;

    if (totalSales >= 1e6)

    {

        percentage = 0.01;

    }

    else if (totalSales >= 5e5)

    {

        percentage = 0.02;

    }

    else if (totalSales >= 1e5)

    {

        percentage = 0.03;

    }

    else if (totalSales >= 1e4)

    {

        percentage = 0.05;

    }

    else

        percentage = 0;

    return percentage;

}

float CalculateTotalComission(float totalSales)

{

    float percent = GetCommissionPercentage(totalSales);

    float totalCommision = totalSales \* percent;

    return totalCommision;

}

int main()

{

    float totalSales = ReadTotalSales();

    cout << "Comission   Percentage : " << GetCommissionPercentage(totalSales) << endl;

    cout << "Total   Percentage : " << CalculateTotalComission(totalSales);

}

#35

struct strPiggyBankContent

{

    int penny;

    int nickel;

    int dime;

    int quarter;

    int dollar;

};

strPiggyBankContent ReadPiggyBankContent()

{

    strPiggyBankContent piggyBankContent;

    cout << "Please enter  pennies  : \n";

    cin >> piggyBankContent.penny;

    cout << "Please enter  nickels  : \n";

    cin >> piggyBankContent.nickel;

    cout << "Please enter  dimes  : \n";

    cin >> piggyBankContent.dime;

    cout << "Please enter  quarter  : \n";

    cin >> piggyBankContent.quarter;

    cout << "Please enter  dollar  : \n";

    cin >> piggyBankContent.dollar;

    return piggyBankContent;

}

int CalculateTotalPennies(strPiggyBankContent piggyBankContent)

{

    const short Penny = 1, Nickel = 5, Dime = 10, Quarter = 25, Dollar = 100;

    int totalPennies = piggyBankContent.penny \* Penny + piggyBankContent.dime \* Dime + piggyBankContent.nickel \* Nickel + piggyBankContent.quarter \* Quarter + piggyBankContent.dollar \* Dollar;

    return totalPennies;

}

int main()

{

    int totalPennies = CalculateTotalPennies(ReadPiggyBankContent());

    float totalDollars = (float)totalPennies / 100;

    cout << "Total Pennies : " << totalPennies << "\n";

    cout << "TotalDollars: " << totalDollars << "\n";

}

#36

enum enOperationType

{

    Multiplication = '\*',

    Subtraction = '-',

    Additin = '+',

    Division = '/'

};

float ReadNumber()

{

    float number;

    cout << "Enter Number ? " << "\n";

    cin >> number;

    return number;

}

enOperationType ReadOpType()

{

    char OT = '\*';

    cout << "Enter Operation  type (\*,-,+,/)? " << "\n";

    cin >> OT;

    return (enOperationType)OT;

}

float getResultFromOperation(float number1, float number2, enOperationType opType)

{

    switch (opType)

    {

    case enOperationType::Multiplication:

        return number1 \* number2;

        break;

    case enOperationType::Subtraction:

        return number1 - number2;

        break;

    case enOperationType::Additin:

        return number1 + number2;

        break;

    case enOperationType::Division:

        return number1 / number2;

        break;

    default:

        return number1 \* number2;

    }

}

int main()

{

    float number1 = ReadNumber();

    float number2 = ReadNumber();

    enOperationType  opType =ReadOpType();

    cout << "Result = "<<  getResultFromOperation(number1 ,number2 ,opType) <<endl;

}

#37

float ReadNumber(string message)

{

    float num;

    cout << message << endl;

    cin >> num;

    return num;

}

float sumNmbers()

{

    int sum = 0, Number = 0, count = 0;

    do

    {

        Number = ReadNumber("please enter number " + to\_string(count));

        if (Number == -99)

            break;

        sum += Number;

        count++;

    } while (Number != -99);

    return sum;

}

int main()

{

    float result = sumNmbers();

    cout << "Result is : " << result;

}

#38

enum enPrimeNotPrime

{

    Prime = 1,

    NotPrime = 2

};

float ReadPositiveNumbers(string message)

{

    float num;

    do

    {

        cout << message << endl;

        cin >> num;

    } while (num <= 0);

    return num;

}

enPrimeNotPrime checkPrime(int number)

{

    int middleOfNum = round(number / 2);

    for (int counter = 2; counter <= middleOfNum; counter++)

    {

        if (number % counter == 0)

        {

            return enPrimeNotPrime ::NotPrime;

        }

    }

    return enPrimeNotPrime::Prime;

}

void PrintNumberType(int number)

{

    switch (checkPrime(number))

    {

    case enPrimeNotPrime::Prime:

        cout << "The number is Prime \n";

        break;

    case enPrimeNotPrime::NotPrime:

        cout << "The number is  Not Prime \n";

        break;

    default:

        break;

    }

}

int main()

{

    float number = ReadPositiveNumbers("Enter a Positive number");

    PrintNumberType(number);

}

#39

float ReadPositiveNumber(string message)

{

    float num;

    do

    {

        cout << message << endl;

        cin >> num;

    } while (num <= 0);

    return num;

}

float calculateReminder(float cashPaid, float totalBill)

{

    return cashPaid - totalBill;

}

void printResuult(float reminder)

{

    cout << "reminder : " << reminder << "\n";

}

int main()

{

    float number;

    float cashPaid = ReadPositiveNumber("Please enter  cash Paid  : ");

    float totalBill = ReadPositiveNumber("Please enter  Total Bill   : ");

    printResuult(calculateReminder(cashPaid, totalBill));

}

#40

float ReadPositiveNumber(string message)

{

    float num;

    do

    {

        cout << message << endl;

        cin >> num;

    } while (num <= 0);

    return num;

}

float TotalBillAfterTaxAndServiceFee(float billValue)

{

    billValue += (billValue \* 0.1);

    billValue += (billValue \* 0.16);

    return billValue;

}

int main()

{

    float billValue = ReadPositiveNumber("Please enter  Bill Value :");

    cout << "Total Bill before is : " << billValue << endl;

    float billValueAfter = TotalBillAfterTaxAndServiceFee(billValue);

    cout << "Total Bill After Service and tax fee , bill will be : " << billValueAfter << "\n";

}

#41

float ReadPositiveNumber(string message)

{

    float num;

    do

    {

        cout << message << endl;

        cin >> num;

    } while (num <= 0);

    return num;

}

float HoursToDays(float totalHours)

{

    return totalHours / 24;

}

float HoursToWeeks(float totalHours)

{

    return totalHours / 168;

}

float DaysToWeeks(float totalDays)

{

    return totalDays / 7;

}

int main()

{

    float numOfHours = ReadPositiveNumber("Please Enter Number Of Hours :");

    float totalDays = HoursToDays(numOfHours);

    float totalWeeks = HoursToWeeks(numOfHours);

    float totalWeeksFromDays = DaysToWeeks(totalDays);

    cout << "total hours : " << numOfHours << endl;

    cout << "total days :" << totalDays << endl;

    cout << "total weeks from hours  :" << totalWeeks << endl;

    cout << "total Weeks form days :  " << totalWeeksFromDays << endl;

}

#42

struct strTaskDuration

{

    float numOfDays, numOfHours, numOfMinutes, numOfSeconds;

};

float ReadPositiveNumber(string message)

{

    float num;

    do

    {

        cout << message << endl;

        cin >> num;

    } while (num <= 0);

    return num;

}

strTaskDuration ReadTaskDuration()

{

    strTaskDuration taskDuration;

    taskDuration.numOfDays = ReadPositiveNumber("Please enter number of days    :");

    taskDuration.numOfHours = ReadPositiveNumber("Please enter number of hours    :");

    taskDuration.numOfMinutes = ReadPositiveNumber("Please enter number of minutes    :");

    taskDuration.numOfSeconds = ReadPositiveNumber("Please enter number of seconds    :");

    return taskDuration;

}

float TaskDurationInSeconds(strTaskDuration taskDuration)

{

    float totalSeconds = 0;

    totalSeconds += taskDuration.numOfDays \* 24 \* 60 \* 60;

    totalSeconds += taskDuration.numOfHours \* 60 \* 60;

    totalSeconds += taskDuration.numOfMinutes \* 60;

    totalSeconds += taskDuration.numOfSeconds;

    return totalSeconds;

}

float totalSeconds(float numOfDays, float numOfHours, float numOfMinutes, float numOfSeconds)

{

}

int main()

{

    float numOfDays, numOfHours, numOfMinutes, numOfSeconds;

    strTaskDuration task = ReadTaskDuration();

    float totalseconds = TaskDurationInSeconds(task);

    cout << "Total Seconds : " << totalseconds;

    return 0;

}

#43

struct strTaskDuration

{

    float numOfDays, numOfHours, numOfMinutes, numOfSeconds;

};

float ReadPositiveNumber(string message)

{

    float num;

    do

    {

        cout << message << endl;

        cin >> num;

    } while (num <= 0);

    return num;

}

strTaskDuration secondsToTaskDuration(float totalseconds)

{

    strTaskDuration taskDuration;

    const float secondsPerDay = 24 \* 60 \* 60;

    const float secondsPerHour = 60 \* 60;

    const float secondsPerMinute = 60;

    float reminder;

    float numOfSeconds;

    taskDuration.numOfDays = floor(totalseconds / secondsPerDay);

    reminder = fmod(totalseconds, secondsPerDay);

    taskDuration.numOfHours = floor(reminder / secondsPerHour);

    reminder = fmod(reminder, secondsPerHour);

    taskDuration.numOfMinutes = floor(reminder / secondsPerMinute);

    reminder = fmod(reminder, secondsPerMinute);

    taskDuration.numOfSeconds = reminder;

    return taskDuration;

}

void printTaskDurationDetails(strTaskDuration taskDuration)

{

    cout << taskDuration.numOfDays << " : " << taskDuration.numOfHours << ": " << taskDuration.numOfMinutes << ": " << taskDuration.numOfSeconds;

}

int main()

{

    float totalSeconds = ReadPositiveNumber("Please enter totalSeconds   : ");

    printTaskDurationDetails(secondsToTaskDuration(totalSeconds));

    return 0;

}

#44

enum enDaysOfWeek

{

    Sun = 1,

    Mon,

    Tue,

    Wed,

    Thu,

    Fri,

    Sat

};

int ReadNumberInRange(string message, int from, int to)

{

    int num;

    do

    {

        cout << message << endl;

        cin >> num;

    } while (num < from || num > to);

    return num;

}

enDaysOfWeek ReadDayOfWeek()

{

    return (enDaysOfWeek)ReadNumberInRange("please enter day number sun=1,mon=2,...,sat=7", 1, 7);

}

string GetDayOfWeek(enDaysOfWeek daysOfWeek)

{

    switch (daysOfWeek)

    {

    case enDaysOfWeek::Sun:

        return "Sunday";

        break;

    case enDaysOfWeek::Mon:

        return "Monday";

        break;

    case enDaysOfWeek::Tue:

        return "Tuesday";

        break;

    case enDaysOfWeek::Wed:

        return "Wednesday";

        break;

    case enDaysOfWeek::Thu:

        return "Thursday";

        break;

    case enDaysOfWeek::Fri:

        return "Friday";

        break;

    case enDaysOfWeek::Sat:

        return "Saturday";

        break;

    }

}

int main()

{

    string day = GetDayOfWeek(ReadDayOfWeek());

    cout << "day is  : " << day << endl;

}