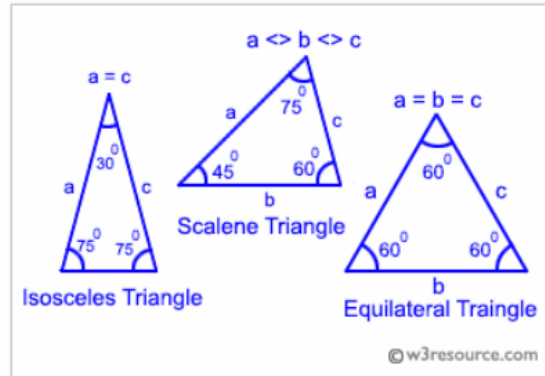


Problem 1: Write a C program to check whether a triangle is Equilateral, Isosceles or Scalene. Equilateral triangle: An equilateral triangle is a triangle in which all three sides are equal. In the familiar Euclidean geometry, equilateral triangles are also equiangular; that is, all three internal angles are also congruent to each other and are each 60° . Isosceles triangle: An isosceles triangle is a triangle that has two sides of equal length. Scalene triangle: A scalene triangle is a triangle that has three unequal sides, such as those illustrated above.



PL HW4
111261012
張晉瑜

Sample Output:

```
Input three sides of triangle: 50 50 60
This is an isosceles triangle.
```

Problem 1 :

Code

```
1. #include <iostream>
2. #include <map>
3. using namespace std;
4. int main(){
5.     int a[3];
6.     /*原本想優化 IO*/
7.     //cin.tie(0);
8.     //ios_base::sync_with_stdio(false);
9.     cout<<"Input three sides of triangle: ";
10.    while(cin>>a[0]>>a[1]>>a[2]){
11.        map<int,int> ans;
12.        for(int i=0;i<3;i++) ans[a[i]]++; //利用 STL 的 map 當作 hash table
13.        cout<<"This is an ";
14.        switch(ans.size()){ //ans 的 size 就是邊長種類的數量
15.            case 1:
16.                cout<<"Equilateral ";
17.                break;
18.            case 2:
19.                cout<<"Isosceles ";
20.                break;
21.            case 3:
22.                cout<<"Scalene ";
23.                break;
24.        }
25.        cout<<"Triangle\nInput three sides of triangle: ";
26.    }
27. }
```

Output

```
Input three sides of triangle: 1 1 1
This is an Equilateral Triangle
Input three sides of triangle: 1 2 1
This is an Isosceles Triangle
Input three sides of triangle: 1 2 3
This is an Scalene Triangle
Input three sides of triangle:
```

Problem 2: Please write a program that reads two numbers X, Y, and then calculates their sum ($X + Y$), subtraction ($X - Y$), multiplication ($X * Y$). Meanwhile, you are requested to use the repetition structure for the same calculation of the input of (X, Y), (X + 1, Y + 1),, and (X + 4, Y + 4).

Example:
Input X and Y as 1 and 2, respectively. The output in the below.

1 2					
X	Y	X+Y	X-Y	X*Y	
1	2	3	-1	2	
2	3	5	-1	6	
3	4	7	-1	12	
4	5	9	-1	20	
5	6	11	-1	30	

Problem 3: Please write a program to print the conversion table of Celsius-Fahrenheit. Before printing the table, the program first asks the user input “lower”, “upper” and “step” values. When a user enters Celsius, your program will answer its corresponding Fahrenheit.

The resulting output would look like this:

Celsius	Fahrenheit
0	32.00
1	33.80
2	35.60
3	37.40

Problem 2 :

Code

```
1. #include <iostream>
2. using namespace std;
3.
4. int main(){
5.     int x,y;
6.     cin>>x>>y; //input
7.     cout<<"X\tY\tX+Y\tX-Y\tX*Y\n";
8.     for(int i=0;i<4;i++){
9.         for(int j=0;j<9;j++)
10.             cout<<"-";
11.     }
12.     cout<<endl;
13.     for(int i=x;i<=x+4;i++){
14.         printf("%d\t%d\t%d\t%d\t%d\n",i,y,i+y,i-y,i*y);
15.         y++;
16.     }
17. }
18.
```

Output

3 4				
X	Y	X+Y	X-Y	X*Y
3	4	7	-1	12
4	5	9	-1	20
5	6	11	-1	30
6	7	13	-1	42
7	8	15	-1	56

Problem 3 :

Code

```
1. #include <iostream>
2. #include <vector>
3. #include <algorithm>
4. using namespace std;
5.
6. bool upper(int a,int b){//sort by upper
7.     return a<b;
8. }
9. bool lower(int a,int b){          //sort by lower
10.    return a>b;
11. }
12.
13. int main(){
14.     cin.tie(0);
15.     string cmd;//我很抱歉 我不知道 step 是什麼意思
16.     int tem,n=4;//tem 用來輸入，然後丟進去 vector
17.     vector<int> v;
18.     cin>>cmd;
19.     while(n--&&cin>>tem){ //理論上可以輸入至 ctrl+z(結束)，但這邊用 4 個資料來演示
20.         v.push_back(tem);
21.     }
22.     if(cmd == "lower")          //我還是很抱歉 我看不懂 step 或是我就在瞎搞
23.         sort(v.begin(),v.end(),lower);    //STL sort
24.     else if(cmd == "upper")
25.         sort(v.begin(),v.end(),upper);    //using iterator to operate
26.
27.     printf("Celsius\t\t\tFahrenheit\n");
28.     for(int i=0;i<4;i++){
29.         for(int j=0;j<9;j++)
30.             cout<<"-";
31.     }
32.     for(vector<int>::iterator it = v.begin();it!=v.end();it++)
33.         printf("\n%d\t\t\t%.2f",*it,(*it)*1.8+32.0);
34. }
35.
```

Output

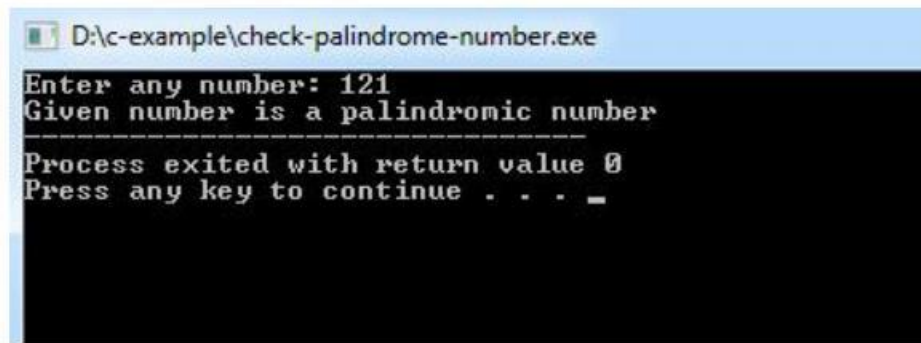
lower 0 1 2 3		
Celsius		Fahrenheit
3		37.40
2		35.60
1		33.80
0		32.00

upper 3 2 1 0		
Celsius		Fahrenheit
0		32.00
1		33.80
2		35.60
3		37.40

step 0 3 1 2		
Celsius		Fahrenheit
0		32.00
3		37.40
1		33.80
2		35.60

Problem 4: Please write a C program for the palindrome number check. For example, the given number = 213312 equals the reverse of the number.

Program output:



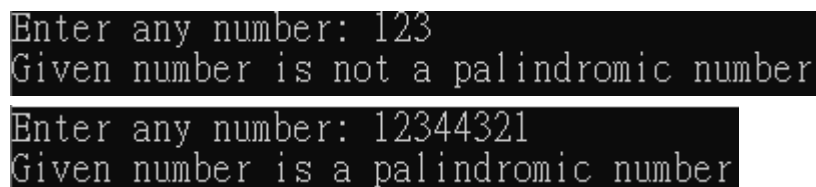
```
D:\c-example\check-palindrome-number.exe
Enter any number: 121
Given number is a palindromic number
-----
Process exited with return value 0
Press any key to continue . . . _
```

Problem 3 :

Code

```
1. #include <iostream>
2. #include <algorithm>
3. using namespace std;
4.
5. int main(){
6.     string num,renum;
7.     cout<<"Enter any number: ";
8.     cin>>num;
9.     renum = num;           //將 num 複製給 renum
10.    reverse(renum.begin(),renum.end());    //將 renum 反轉，此時 renum 即 num 的反轉
11.    if(num == renum)        //一個字串反轉後跟原來一樣，那就是迴文
12.        cout<<"Given number is a palindromic number";
13.    else
14.        cout<<"Given number is not a palindromic number";
15. }
16.
```

Output



```
Enter any number: 123
Given number is not a palindromic number

Enter any number: 12344321
Given number is a palindromic number
```

Problem 5: Please implement the approximations of Exponential and Natural Logarithm functions in C using Taylor Series expansion.

(1) The exponential function e^x (with base e) has Maclaurin series as follows:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

It converges for all x .

(2) The *natural logarithm* (with base e) has Maclaurin series as follows:

$$\ln(1-x) = -\sum_{n=1}^{\infty} \frac{x^n}{n} = -x - \frac{x^2}{2} - \frac{x^3}{3} - \dots,$$

$$\ln(1+x) = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{x^n}{n} = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots.$$

They converge for $x \in (-1, 1)$. In addition, the series for $\ln(1-x)$ converges for $x = -1$, and the series for $\ln(1+x)$ converges for $x = 1$.

write a C/C++ program that takes as the input value of x (e.g. 0.01) for the convergence, and compute these two functions using above approximations. The programs should sum up n terms, and then stop when the value of the n^{th} term is less than a predefined constant accuracy, set to 0.00001.

Problem 3 :

Code

```
1. #include <iostream>
2. using namespace std;
3. const float minn = 0.00001;
4. int dp[10];
5. int DP(int i){
6.     if(dp[i]) return dp[i];
7.     else{
8.         dp[i] = DP(i-1)*i;
9.         return dp[i];
10.    }
11. }
12. float pow(float x,int y){
13.     float ans=1;
14.     for(int i=0;i<y;i++){
15.         ans*=x;
16.     }
17.     return ans;
18. }
19. int main(){
20.     for(int i=0;i<10;i++) dp[i]=0; //reset the array
21.     dp[0]=1;
22.     dp[1]=1;
23.     float x,e=1,lmn=0,lnp=0,tem=1; //e is e^x(and start to 1),lmn is ln(1-x),lnp is ln(1+x)
24.     cin>>x;
25.     int i=1; //e's Maclaurin series start to 1
26.     while(tem>minn){
27.         tem = pow(x,i)/DP(i);
28.         e+=tem;
29.         i++;
30.     }
31.     cout<<"e^"<<x<<" \t\t= "<<e<<endl;
32.     tem = 1,i = 1; //avoid stoping program
33.     while(tem>minn){
34.         tem = pow(x,i)/i;
35.         lmn-=tem;
36.         i++;
37.     }
38.     cout<<"ln(1-"<<x<<") \t= "<<lmn<<endl;
39.     tem = 1,i = 1;
40.     while(tem>minn){
41.         if((i+1)&1) //(i+1) and 1 等價 (i+1) mod 2
42.             tem = (-1)*pow(x,i)/i;
43.         else
44.             tem = pow(x,i)/i;
45.         lnp+=tem;
46.         i++;
47.         tem = tem>0? tem:(-1)*tem;
48.     }
49.     cout<<"ln(1+"<<x<<")\t= "<<lnp;
50. }
51.
```

Output

0.01		0.5	
e^0.01	= 1.01005	e^0.5	= 1.64872
ln(1-0.01)	= -0.0100503	ln(1-0.5)	= -0.693139
ln(1+0.01)	= 0.00995033	ln(1+0.5)	= 0.405468