//实验1

#include <iostream>

using namespace std;

const float pi = 3.14159;

int main()

{

int i;

float radius,a,b,area;

cout<<" :图形类型如下 ：1.圆，2.长方形，3.正方形";

cin>>i;

switch(i)

{

case 1:

cout<<"enter the radius";

cin>>radius;

area = pi \* radius \* radius;

cout<<"S = "<<area<<endl;

break;

case 2:

cout<<"Enter the length of the rectangle";

cin>>a;

cout<<"Enter the width of the rectangle";

cin>>b;

area = a \* b;

cout<<"S = "<<area<<endl;

break;

case 3:

cout<<"Enter the side length of the square";

cin>>a;

area = a \* a;

cout<<"S = "<<area<<endl;

break;

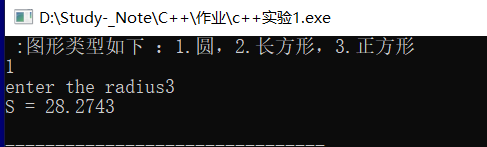
default:

cout<<"不在可选择范围内，请重新输入"<<endl;

}

return 0;

}



//1.2

#include<iostream>

using namespace std;

int main()

{

int sum,i = 1;

do{

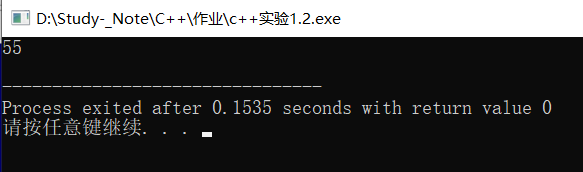
sum += i;

i++;

}while(i != 11);

cout<<sum<<endl;

}



//1.3

#include<iostream>

using namespace std;

int main()

{

int i,sum = 0;

for(i = 1;i < 11; i++)

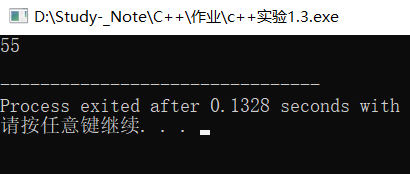
{

sum += i;

}

cout<<sum<<endl;

}



//2.1

#include<iostream>

using namespace std;

float swap(float F)

{

float C;

C = (F - 32) \* 5 / 9.0;

return C;

}

int main()

{

float f,c;

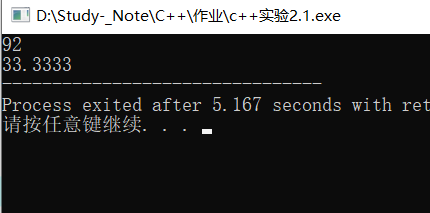
cin>>f;

c = swap(f);

cout<<c;

return 0;

}



//2.2

#include<iostream>

using namespace std;

int max1(int a = 0,int b = 0)

{

return a>b?a:b;

}

int max1(int a,int b,int c)

{

int tmp = max1(a,b);

return tmp>c?tmp:c;

}

double max1(double a = 0,double b = 0)

{

return a>b?a:b;

}

double max1(double a,double b,double c)

{

int tmp = max1(a,b);

return tmp>c?tmp:c;

}

int main()

{

int n1,n2,n3;

cout<<"please enter two :";

cin>>n1>>n2;

cout<<"the max is"<<max1(n1,n2)<<endl;

cout<<"please enter three :";

cin>>n1>>n2>>n3;

cout<<"the max is"<<max1(n1,n2,n3)<<endl;

double b1,b2,b3;

cout<<"please enter two :";

cin>>b1>>b2;

cout<<"the max is"<<max1(b1,b2)<<endl;

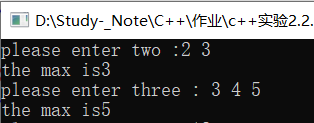
cout<<"please enter three :";

cin>>b1>>b2>>b3;

cout<<"the max is"<<max1(b1,b2,b3)<<endl;

return 0;

}



#include<iostream>

#include<math.h>

using namespace std;

int main()

{

double x,y;

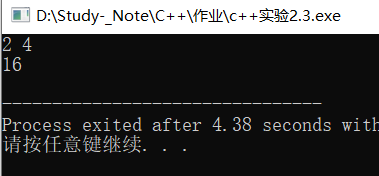
cin>>x>>y;

x = pow(x,y);

cout<<x<<endl;

return 0;

}



//实验三

#include<iostream>

using namespace std;

class TDate{

public:

TDate()

{

year = 2018;month = 1;day = 1;

cout<<".调用缺省的构造函数"<<endl;

}

TDate(int y,int m = 1,int d = 1)

{

year = y;month = m;day = d;

cout<<".调用带参数的构造函数"<<endl;

}

void SetDate(int y,int m,int d);

void AddOneDay();

void Show();

~TDate(){cout<<".调用析构函数"<<endl; }

private:

int year;

int month;

int day;

};

int MonthMaxDay(int year,int month)

{

int days;

switch(month)

{

case 1:

case 3:

case 7:

case 8:

case 10:

case 12:days = 31;break;

case 2:if(year % 4 == 0 && year % 100 != 0||year % 400 == 0) days = 29;else days = 28;break;

default:days = 30;break;

}

return days;

}

void TDate::Show()

{

cout<<year<<".年"<<month<<".月"<<day<<".日"<<endl;

}

void TDate::SetDate(int y,int m,int d)

{

year = y;

month = m;

day = d;

}

void TDate::AddOneDay()

{

day = day + 1;

if(day > MonthMaxDay(year,month))

{

month = month + 1;

day = 1;

}

if(month > 12)

{

year = year + 1;

month = 1;

}

}

int main()

{

TDate d;

cout<<".不带参数的对象: ";

d.Show();

TDate d1(2019);

cout<<".构造函数重载";

d1.Show();

TDate d2(2019,2,28);

cout<<".带两个参数";

d2.Show();

TDate d3(2019,4,02);

cout<<".三个参数";

d3.Show();

d3.SetDate(2019,2,28);

cout<<".重新设置日期";

d3.Show();

cout<<".增加一天后";

d3.AddOneDay();

d3.Show();

return 0;

}

