# 信息学课堂笔记11

# 一、课堂习题

# [B2142]求 1+2+3+...+N 的值

## 题目描述

用递归的方法求 1+2+3+ ..... +N 的值。

## 输入格式

输入 N。

## 输出格式

输出和。

## 样例 #1

### 样例输入 #1

```
5
```

### 样例输出 #1

```
15
```

## 提示

N < 200

```cpp
#include<bits/stdc++.h>
using namespace std;
int n, sum = 0;
void f(int a){
    if (a > n) {   //如果a已经超过了我们要计算的上限，此时就不该继续做递归了
        return ;
    } else {
        sum += a; //sum 加上当前的值a
        f(a + 1); //让函数继续计算下一个值a + 1
    }
}
int main(){
    cin >> n;
    f(0);
    cout << sum;
    return 0;
}
```

# 同理，求 1×2×3×...×N 的值

与上一份代码所不同，直接修改➕，有两个坑，如注释所示

```
#include<bits/stdc++.h>
using namespace std;
int n, sum = 1;                    //sum=1

void f(int a){
    if (a > n){
        return ;
    } else {
        sum *= a;
        f(a + 1);
    }
}

int main(){
    cin >> n;
    f(1);                         //f(1)
    cout << sum;
    return 0;
}
```

# 二、期末考试测试卷讲解

## 一．单项选择题

1.2019年10月14日是星期一，2020年10月14日是（A）

A.星期三　B.星期五　C.星期一　D.星期六

**解析：计算出两个日期的差值，用差值%7得到A，然后在前者的星期基础上面加上A天，或者在后者的星期上减上A天。**

2.一只蜗牛在10米深的井底向上爬，每小时向上爬3米后要下降2米，这只蜗牛要几个小时才能爬出井口（B）

A.7　B.8　C.9　D.10

**解析：因为最后一小时肯定是只有往上爬的过程，没有下降的过程，所以10-3，之前的每一小时，其实只往上爬了1米（3-2），所以需要（10-3）/1+1=8**

3. 以下不适合定义为用户标识符的是（A）

A. double　B. abc　C. Source　D. PI

**解析：double是变量类型**

4.题目给你一个整数m，保证m<100，此时程序中该如何定义变量m（D）

A.long long m(m<100);

B.int m(m<100);

C.int m(<100);

D.int m;

**解析：类型 变量名**

5. 在C++语言中，字符型数据在内存中以（A）形式存放。

A. ASCII码

B. BCD码

C. 反码

D. 国标码

**解析：char以ASCII码值存放**

6. 设a和b均为int型变量，且a=5.5、b=2.5，则表达式a+b/b的值是(D)

A. 6.500000

B. 6.000000

C. 5.500000

D. 6

**解析：a，b都为整型：a+b/b=5+2/2=6，**

7.若希望当A的值为奇数时，表达式的值为”true”, A的值为偶数时，表达式的值为”false”。则以下不能满足要求的表达式是（A、C）。

A. A%21　B. !(A%20)　C. !(A%2)　D .A%2

**解析：A：当A为负奇数时，表达式为false**

**C：”！“为取反的意思，所以A为奇数时，A%2=1，为真，再取反，为假，所以不能满足。**

8.阅读以下程序,当输入数据的形式为”25,13,10”正确的输出结果为（ A ）

```
#include<iostream>
using namespace std;
int main()
{
    int x,y,z;
    cin>>x>>y>>z;
    cout<<"x+y+z="<<x+y+z;
}
```

A x+y+z=48　B x+y+z=35

C x+y+z=47 D不确定值

**解析：输出为x+y+z=48，**

9.小明在做一道题的过程中，发现待输入的数据为有如下内容：

21435674 5678123451 2.7 c hello

小明要想将所有的数据正常存储，应该按顺序使用以下（ A ）变量类型

A int , long long, double , char , string

B int, int, char, char

C int , string, double, char, char

D.int, string, string, string

10.若变量a是int类型，并执行了语句：a='A'+1.6；，则正确的叙述是（D ）

A.不允许字符型和浮点型相加

B. a的值是字符型

C. a的值是浮点型

D. a的值还是整型

**解析：‘A’在参与运算时，会转化为ASCII码值，同时，a是int类型，所以到最后，值还是整型。**

11.判断char型变量ch是否为大写字母的正确表达式是_**C**。

A. ‘A’<=ch<=‘Z’　　B.(ch> = ‘A’)&(ch<=‘Z’)

C.(ch>=‘A’)&&(ch<=‘Z’)　D.(‘A’< = ch)AND(‘Z’> = ch)

**解析：A，数学表达式；B，语法错误；D，C++中没有AND**

12. 下列说法中***错误***的是(C)。

A 构成数组的所有元素的数据类型必须是相同的。

B 一维数组的下标从0开始。

C 一维数组元素的下标从1开始。

D int a[3+5]; 是合法的。

**解析：一维数组的元素下标从0开始**

13.执行以下程序的输出结果是（C）

```
int main()
{
  int i,s=0;
  for(i=1;i<10;i+=2)          //解析：i从1开始，每次变大2
    {
      s+=i;                   //s=1+3+5+7+9
    }
    cout<<s;
}
```

A)自然数1~9的累加和　B) 自然数1~10的累加和

C)自然数1~9中奇数之和　D) 自然数1~10中偶数之和

14. 有以下程序段

```
int  k=0;
while(k)
{
    k++;
}
```

则while循环体执行的次数是( C )

A) 无限次　　B) 有语法错，不能执行

C) 一次也不执行　　D) 执行1次

**解析：k=0,为假，不进入循环**

15. 若i、j已定义为int类型，则以下程序段运行结果有（D）个“*”

```
for(i=5;i>1;i--)
{
    for(j=0;j<4;j++)
    {
        cout<<'*';
    }
}
```

A)5　B) 4　C)20　D)16

**解析：双层for循环，内外两层都是循环4次，则*的总数量为4×4=16**

# 二、填空题

1. 数据类型：整数型（ int ）长整数型（ long long ）小数型（ double ）布尔型（ bool ）字符型（ char ）。
2. 布尔真值（ true ）布尔假值（ false ）。
3. 循环的跳过和中断直接进入当前层循环的下一次循环（ continue ） 跳出当前层循环（ break ）
4. 要想使用pow()函数，需要添加（ math.h或cmath ）头文件。

# 三、阅读程序

1.

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

int main()
{
    int a=12, b= -34, c=56, min;
    min=a;
    if(min>b)                                   //不难发现此程序是求出a、b、c中的最小值
    {
        min=b;
    }
    if(min>c)
    {
        min=c;
    }
    cout<<"min="<<min;
    return 0;
}
```

**运行结果为：_min=-34**

2.

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

int main() {
    int  n=9;
    while(n>6)
    {
        n--;
        cout<<n;
    }
    return 0;
}
```

**运行结果为：__876**

三．程序填空

1.交换a,b变量的值

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

int main(){
    int a=5,b=2,c;
    __c=a__;
    __a=b___;
    __b=c_;
    cout<<a<<" "<<b;
    return 0;
}

//或:
//c=b;
//b=a;
//a=c;

//或:
//a+=b;
//b=a-b;
//a-=b;
```

2.求0～999之间的所有“水仙花数”。

“水仙花数”是指一个三位数，其各位数字的立方和正好好等于该数本身。

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

int main()
{
    for(int i=100;i<=999;i++)
    {
        int a = 0; / /个位
        int b = 0; //十位
        int c = 0; //百位
        a =___i%10___;
        b =___i/10%10____;
```

```
        c =___i/100___;                                    //如果i大于4位数，则需还要%10;
        if(i == a*a*a + b*b*b + c*c*c)
        {
            cout << i << endl;
        }
  }
  return 0;
}
```

# 三、课堂笔记

## 1、i++与++i的区别

简单理解：

i++返回i原本的值，而++i返回i+1后的值

参考代码：

```
#include<bits/stdc++.h>
using namespace std;

int main(){
    int i=0;
    cout<<i++<<endl;
    cout<<++i<<endl;
    return 0;
}
```

结果：

解析：第一个cout输出i的初始值0，但是在这一句语句执行完之后，i的值变为1

第二个cout输出2，因为，返回的值为(i+1)的值，2

## 2、动态数组vector（拓展知识）

头文件：

优点：无法知道自己需要的数组的规模多大时,用其来解决问题可以达到最大节约空间的目的.

缺点：消耗的时间太多，浪费在开辟新空间，转移数据上面。

### 1、创建

```
    vector<元素类型> 变量名;
    vector<int> v;          //声明一个int类型的向量v
    要注意的是，通过这种方式创建的vector是一个空的vector，长度为0的。所以，访问l[0]都是错误的，会导致程序RE。
    那我们希望像创建静态数组例如a[100000]这样创建一个vector该怎么办呢？
    需要使用vector内置自带的 构造函数。
    vector<int> a(5); 这表示创建了一个int类型的动态数组a，初始长度为5，具体范围为a[0] ~ a[4];
```

```
    还有没有别的方式可以创建一个带初始长度的vector呢？

    int n;
    cin >> n;
    vector<int> nums(n); //在定量读取n个整数，并开辟n个长度的数组时，需要这样使用，如果同学你比较习惯[1,n]的数组范围，那你可以开辟到nums(n + 1);

    int n;
    cin >> n;
    vector<int> nums(n, 3); //这是带值默认构造函数
    //通过这种方式创建的vector，长度为n，其中，每一位的初始值都设置为3.

    int n;
    vector<int> a = {1, 3, 2, 4, 5}; //这样表示创建了一个带有初始化值的vector，长度为5，每一位的值都是大括号里分别每一位对应的值

    vector<int> a = {1, 3, 2, 4, 5};
    vector<int> b(a); //这是拷贝构造创建，我们创建的b数组，直接获取到a数组的长度，a数组的值，直接完完整整的将a数组复制到了b中。
```

## 2、插入函数

void push_back( val );

添加值为val的元素到当前vector末尾

```
v.push_back(val);


需要注意的是，vector还有一个函数叫做emplace_back，他的用法和push_back一模一样，但是在数据量极大极大的时候，emplace_back会略优于push_back。这涉及到C++的浅拷贝，深拷贝，写时拷贝的知识，大家不需要对此做了解，只需要记住结论即可。

v.emplace_back(val);
v.push_back(val);
```

## 3、大小函数

size() 返回Vector元素数量的大小

```
v.size();
这个函数的作用同
v.length();
要注意的是，这个函数返回的值是一个size_t类型，再精确来说，它是一个unsigned int，无符号整形的值
什么叫做无符号的整形，也就是这个整型变量没有负数，只有正数，它的二进制第一位符号位取消了，直接存值，所以从无符号整数转变成有符号整数最好能够明确一下。
比如说在下面这个代码
    for (int i = 0; i < a.size(); i++) {
        sum += a[i];
    }
同学们观察一下 i < a.size()这句话，小于号的左边是个int类型，小于号的右边是个unsigned int类型，这不是同类型运算，这样写不好！
那应该怎么写更好呢？
    int n = a.size();
    for (int i = 0; i < n; i++) {
```

```
        sum += a[i];
    }
```
我们在循环外部的时候，先把无符号整形转变成有符号的int，再在循环判断的时候就是同类型判断了。

## 4、遍历动态数组（与静态数组一样）

```
for(int i = 0; i < v.size(); i++) {
    cout << v[i] << " ";
}


在C++14版本，C++还为我们提供了一种，不带循环变量的循环方式，以下内容不需要同学们掌握！！！！
int n;
cin >> n;
vector<int> nums(n);
for (auto& x : nums) {
    cin >> x;
}
for (auto& x : nums) {
    cout << x << " ";
}

在先前我们学习遍历的时候，各位同学都知道，我们应该使用带有循环变量的i，然后通过访问nums[i]的形式来访问数组里的每一个值。
但是C14新特性中，为我们提供了一种让变量自动遍历，直接访问到数组里具体某个值的方法，比如这里的x
auto& x 就表示我们让编译器自动推导一个类型，这里应该是int，让这个x一直代替数组里的每一个值nums[0], nums[1] ...
这样的话，我们在cin的时候对x来cin即可，输出的时候输出x就行。
那这样写的好处是什么呢？代码精简，很简短，可读性提高了。是一种类Python的写法。
这样写的缺点是什么呢？大家可以看一下下面这句代码：
for (int i = 0; i < n - 1; i++) {
    sum[i] = sum[i] + sum[i + 1];
}
vector<int> nums(n);
for (int i = 0; i < n; i++) {
    nums[i] = i;
}
在第一个例子中，我们需要访问到i + 1这个变量，在第二个例子中，我们需要直接保存循环变量这个标号，那我们在需要用到循环变量这个具体的标号的时候，就不能使用自动推导遍历了，那当我们不关心循环变量的作用的时候，就可以使用auto& x : 这种自动推导形式的遍历来减少我们的代码量
```

## 5、测试代码

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

int main(){
    vector<int> v;              //声明动态数组，v
    int n;
    cin >> n;
    while (n--){                //给数组输入n个值
        int a;
        cin >> a;
```

```
        v.push_back(a);
    }
    cout << "数组大小：" << v.size() << endl;  //数组的大小
    cout << "数组的内容:" << endl;
    for(int i = 0; i < v.size(); i++) {          //遍历数组
        cout << v[i] << " ";
    }
    cout << endl;
    return 0;
}
```

## 6、动态数组的储存

声明动态数组的内部结构

储存了五个元素的动态数组的内部结构

## 7、string与vector的对比

储存结构十分相似：

string可以简单的理解为字符数组（不太严谨）

# 四、课后作业

洛谷上面的三道题（用动态数组实现）