# 6-1 简单输出整数（10 分）

本题要求实现一个函数，对给定的正整数N，打印从1到N的全部正整数。

### 函数接口定义：

void PrintN ( int N );

其中N是用户传入的参数。该函数必须将从1到N的全部正整数顺序打印出来，每个数字占1行。

### 裁判测试程序样例：

#include <stdio.h>

void PrintN ( int N );

int main ()

{

int N;

scanf("%d", &N);

PrintN( N );

return 0;

}

/\* 你的代码将被嵌在这里 \*/

### 输入样例：

3

### 输出样例：

1

2

3

# 6-2 多项式求值（15 分）

本题要求实现一个函数，计算阶数为n，系数为a[0] ... a[n]的多项式*f*(*x*)=∑​*i*=0​*n*​​(*a*[*i*]×*x*​*i*​​) 在x点的值。

### 函数接口定义：

double f( int n, double a[], double x );

其中n是多项式的阶数，a[]中存储系数，x是给定点。函数须返回多项式f(x)的值。

### 裁判测试程序样例：

#include <stdio.h>

#define MAXN 10

double f( int n, double a[], double x );

int main()

{

int n, i;

double a[MAXN], x;

scanf("%d %lf", &n, &x);

for ( i=0; i<=n; i++ )

scanf(“%lf”, &a[i]);

printf("%.1f\n", f(n, a, x));

return 0;

}

/\* 你的代码将被嵌在这里 \*/

### 输入样例：

2 1.1

1 2.5 -38.7

### 输出样例：

-43.1

# 6-3 简单求和（10 分）

本题要求实现一个函数，求给定的N个整数的和。

### 函数接口定义：

int Sum ( int List[], int N );

其中给定整数存放在数组List[]中，正整数N是数组元素个数。该函数须返回N个List[]元素的和。

### 裁判测试程序样例：

#include <stdio.h>

#define MAXN 10

int Sum ( int List[], int N );

int main ()

{

int List[MAXN], N, i;

scanf("%d", &N);

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

scanf("%d", &List[i]);

printf("%d\n", Sum(List, N));

return 0;

}

/\* 你的代码将被嵌在这里 \*/

### 输入样例：

3

12 34 -5

### 输出样例：

41

# 6-4 求自定类型元素的平均（10 分）

本题要求实现一个函数，求N个集合元素S[]的平均值，其中集合元素的类型为自定义的ElementType。

### 函数接口定义：

ElementType Average( ElementType S[], int N );

其中给定集合元素存放在数组S[]中，正整数N是数组元素个数。该函数须返回N个S[]元素的平均值，其值也必须是ElementType类型。

### 裁判测试程序样例：

#include <stdio.h>

#define MAXN 10

typedef float ElementType;

ElementType Average( ElementType S[], int N );

int main ()

{

ElementType S[MAXN];

int N, i;

scanf("%d", &N);

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

scanf("%f", &S[i]);

printf("%.2f\n", Average(S, N));

return 0;

}

/\* 你的代码将被嵌在这里 \*/

### 输入样例：

3

12.3 34 -5

### 输出样例：

13.77

# 6-5 求自定类型元素的最大值（10 分）

本题要求实现一个函数，求N个集合元素S[]中的最大值，其中集合元素的类型为自定义的ElementType。

### 函数接口定义：

ElementType Max( ElementType S[], int N );

其中给定集合元素存放在数组S[]中，正整数N是数组元素个数。该函数须返回N个S[]元素中的最大值，其值也必须是ElementType类型。

### 裁判测试程序样例：

#include <stdio.h>

#define MAXN 10

typedef float ElementType;

ElementType Max( ElementType S[], int N );

int main ()

{

ElementType S[MAXN];

int N, i;

scanf("%d", &N);

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

scanf("%f", &S[i]);

printf("%.2f\n", Max(S, N));

return 0;

}

/\* 你的代码将被嵌在这里 \*/

### 输入样例：

3

12.3 34 -5

### 输出样例：

34.00

作者: 陈越

单位: 浙江大学

时间限制: 400ms

内存限制: 64MB

代码长度限制: 16KB

# 6-6 求单链表结点的阶乘和（15 分）

本题要求实现一个函数，求单链表L结点的阶乘和。这里默认所有结点的值非负，且题目保证结果在int范围内。

### 函数接口定义：

int FactorialSum( List L );

其中单链表List的定义如下：

typedef struct Node \*PtrToNode;

struct Node {

int Data; /\* 存储结点数据 \*/

PtrToNode Next; /\* 指向下一个结点的指针 \*/

};

typedef PtrToNode List; /\* 定义单链表类型 \*/

### 裁判测试程序样例：

#include <stdio.h>

#include <stdlib.h>

typedef struct Node \*PtrToNode;

struct Node {

int Data; /\* 存储结点数据 \*/

PtrToNode Next; /\* 指向下一个结点的指针 \*/

};

typedef PtrToNode List; /\* 定义单链表类型 \*/

int FactorialSum( List L );

int main()

{

int N, i;

List L, p;

scanf("%d", &N);

L = NULL;

for ( i=0; i<N; i++ ) {

p = (List)malloc(sizeof(struct Node));

scanf("%d", &p->Data);

p->Next = L; L = p;

}

printf("%d\n", FactorialSum(L));

return 0;

}

/\* 你的代码将被嵌在这里 \*/

### 输入样例：

3

5 3 6

### 输出样例：

846

# 6-7 统计某类完全平方数（20 分）

本题要求实现一个函数，判断任一给定整数N是否满足条件：它是完全平方数，又至少有两位数字相同，如144、676等。

### 函数接口定义：

int IsTheNumber ( const int N );

其中N是用户传入的参数。如果N满足条件，则该函数必须返回1，否则返回0。

### 裁判测试程序样例：

#include <stdio.h>

#include <math.h>

int IsTheNumber ( const int N );

int main()

{

int n1, n2, i, cnt;

scanf("%d %d", &n1, &n2);

cnt = 0;

for ( i=n1; i<=n2; i++ ) {

if ( IsTheNumber(i) )

cnt++;

}

printf("cnt = %d\n", cnt);

return 0;

}

/\* 你的代码将被嵌在这里 \*/

### 输入样例：

105 500

### 输出样例：

cnt = 6

# 6-8 简单阶乘计算（10 分）

本题要求实现一个计算非负整数阶乘的简单函数。

### 函数接口定义：

int Factorial( const int N );

其中N是用户传入的参数，其值不超过12。如果N是非负整数，则该函数必须返回N的阶乘，否则返回0。

### 裁判测试程序样例：

#include <stdio.h>

int Factorial( const int N );

int main()

{

int N, NF;

scanf("%d", &N);

NF = Factorial(N);

if (NF) printf("%d! = %d\n", N, NF);

else printf("Invalid input\n");

return 0;

}

/\* 你的代码将被嵌在这里 \*/

### 输入样例：

5

### 输出样例：

5! = 120

# 6-9 统计个位数字（15 分）

本题要求实现一个函数，可统计任一整数中某个位数出现的次数。例如-21252中，2出现了3次，则该函数应该返回3。

### 函数接口定义：

int Count\_Digit ( const int N, const int D );

其中N和D都是用户传入的参数。N的值不超过int的范围；D是[0, 9]区间内的个位数。函数须返回N中D出现的次数。

### 裁判测试程序样例：

#include <stdio.h>

int Count\_Digit ( const int N, const int D );

int main()

{

int N, D;

scanf("%d %d", &N, &D);

printf("%d\n", Count\_Digit(N, D));

return 0;

}

/\* 你的代码将被嵌在这里 \*/

### 输入样例：

-21252 2

### 输出样例：

3

# 6-10 阶乘计算升级版（20 分）

本题要求实现一个打印非负整数阶乘的函数。

### 函数接口定义：

void Print\_Factorial ( const int N );

其中N是用户传入的参数，其值不超过1000。如果N是非负整数，则该函数必须在一行中打印出N!的值，否则打印“Invalid input”。

### 裁判测试程序样例：

#include <stdio.h>

void Print\_Factorial ( const int N );

int main()

{

int N;

scanf("%d", &N);

Print\_Factorial(N);

return 0;

}

/\* 你的代码将被嵌在这里 \*/

### 输入样例：

15

### 输出样例：

1307674368000

# 6-11 求自定类型元素序列的中位数（25 分）

本题要求实现一个函数，求N个集合元素A[]的中位数，即序列中第⌊*N*/2+1⌋大的元素。其中集合元素的类型为自定义的ElementType。

### 函数接口定义：

ElementType Median( ElementType A[], int N );

其中给定集合元素存放在数组A[]中，正整数N是数组元素个数。该函数须返回N个A[]元素的中位数，其值也必须是ElementType类型。

### 裁判测试程序样例：

#include <stdio.h>

#define MAXN 10

typedef float ElementType;

ElementType Median( ElementType A[], int N );

int main ()

{

ElementType A[MAXN];

int N, i;

scanf("%d", &N);

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

scanf("%f", &A[i]);

printf("%.2f\n", Median(A, N));

return 0;

}

/\* 你的代码将被嵌在这里 \*/

### 输入样例：

3

12.3 34 -5

### 输出样例：

12.30

# 6-12 判断奇偶性（10 分）

本题要求实现判断给定整数奇偶性的函数。

### 函数接口定义：

int even( int n );

其中n是用户传入的整型参数。当n为偶数时，函数返回1；n为奇数时返回0。注意：0是偶数。

### 裁判测试程序样例：

#include <stdio.h>

int even( int n );

int main()

{

int n;

scanf("%d", &n);

if (even(n))

printf("%d is even.\n", n);

else

printf("%d is odd.\n", n);

return 0;

}

/\* 你的代码将被嵌在这里 \*/

### 输入样例1：

-6

### 输出样例1：

-6 is even.

### 输入样例2：

5

### 输出样例2：

5 is odd.

# 6-13 折半查找（15 分）

给一个严格递增数列，函数int binSearch(SeqList T, KeyType k)用来二分地查找k在数列中的位置。

函数接口定义：

int binSearch(SeqList T, KeyType k)

其中T是有序表，k是查找的值。

裁判测试程序样例：

#include <iostream>

using namespace std;

#define MAXLEN 50

typedef int KeyType;

typedef struct

{ KeyType key;

} elementType;

typedef struct

{ elementType data[MAXLEN+1];

int len;

} SeqList;

void creat(SeqList &L)

{ int i;

cin>>L.len;

for(i=1;i<=L.len;i++)

cin>>L.data[i].key;

}

int binSearch(SeqList T, KeyType k);

int main ()

{ SeqList L; KeyType k;

creat(L);

cin>>k;

int pos=binSearch(L,k);

if(pos==0) cout<<"NOT FOUND"<<endl;

else cout<<pos<<endl;

return 0;

}

/\* 请在这里填写答案 \*/

输入格式：

第一行输入一个整数n，表示有序表的元素个数，接下来一行n个数字，依次为表内元素值。 然后输入一个要查找的值。

输出格式：

输出这个值在表内的位置，如果没有找到，输出"NOT FOUND"。

输入样例：

5

1 3 5 7 9

7

输出样例：

4

输入样例：

5

1 3 5 7 9

10

输出样例：

NOT FOUND