**实验1:复习**

**姓名\_\_金宣成\_\_\_班级\_\_计科03\_\_\_\_学号\_202203151310\_\_**

* **请阅读此说明：实验1满分100分；做完实验后请按要求将代码和截图贴入该文档。然后将此文档、源代码文件（.cpp）打包上传到学习通。**

1. **数组+函数 基础复习部分：（25分）**

**有如下主函数框架：**

**int main()**

**{ unsigned n;**

**int \*p;**

**while(cin>>n)**

**{**

**p= InitArray(n);**

**//初始化一个规模为n的一维整型数组，并给出随机互异的值进行初始化。**

**Display(p,n); //显示数据的初始化情况**

**SortArray(p,n);//根据每个数包含的数字按个位、十位和百位和组成的新数值进行升序整理；如123对应的新数值为321，12的对应的新数值为210**

**Display(p,n); //显示排序后的结果**

**DestroyArray(p); //根据需要使用**

**}**

**return 0;**

**}**

* **实验要求：**

**❶根据主函数工作要求，设计对应的函数原型并将他们实现；**

**❷将所有实现的函数与主函数进行装配测试，并补充完全所有程序运行需要的内容。**

**❸给出测试的截图。分别使用n为12； n为123；n为1234的数据规模**

* **实验提交：**

**将完整源代码+适当注释和测试截图 黏贴在下面。**

* + **代码黏贴处：**

#include <iostream>

#include <string>

#include <iomanip>

#include <cmath>

#include <algorithm>

using namespace std;

bool contain(int num,int \*a,int n){

    for(int i = 0;i < n;i++){

        if(num == a[i]){

            return true;

        }

    }

    return false;

}

void Display(int \* b,int n){

    for(int i = 0;i < n;i++){

        cout << b[i] << ' ';

    }

    cout << endl;

}

int newnum(int n){

    int sum = 0;

    for(int i = 0;i < 3;i++){

        sum = sum \* 10 + n%10;

        n/=10;

    }

    return sum;

}

void SortArray(int \*p, int n){

    for(int i = 1;i < n ;i++){

        for(int j = 0 ;j < n - i ;j++){

            if(newnum(p[j]) > newnum(p[j+1])){

                swap(p[j],p[j+1]);

            }

        }

    }

}

void DestroyArray(int \*p){

    delete [] p;

}

int \* InitArray(int n){

    srand(time(NULL));

    int count = 0;

    int \*a = new int [n];

    while(count < n){

        int num = rand();

        if(!contain(num,a,count)){

            a[count++] = num;

        }

    }

    return a;

}

int main(){

    int \* p;

    int n ;

    while(cin >> n){

        p= InitArray(n);

        Display(p,n);

        SortArray(p,n);

        Display(p,n);

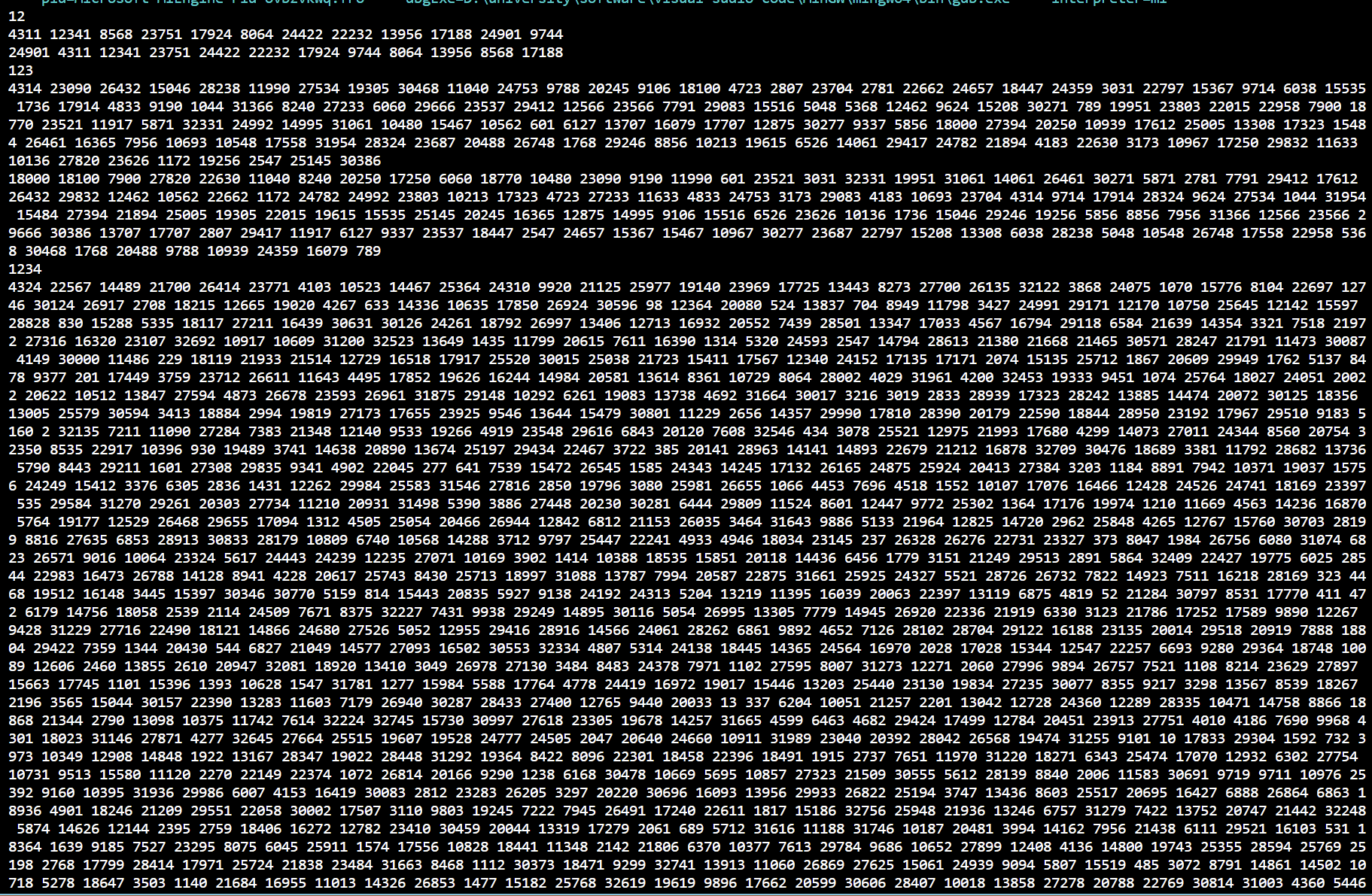
        DestroyArray(p);

    }

    return 0;

}

* + **截图黏贴处：**

****

1. **数组+函数 提高复习部分：（25分）**

**利用1的处理模型，考虑以下测试需求，即程序可能接收多组测试，每组测试由组数，每组的数据个数，每组数组值 构成；最后输出对当前组升序排列的结果。**

**输入测试样本：**

**3**

**3 122 131 133**

**5 4210 2010 3241 1623 232**

**4 235 14 368 201**

**测试输出样本：**

**131 122 133**

**2010 4210 3241 232 1623**

**201 14 235 368**

* **实验要求：**

**❶ 以题1的实现为基础，适当调整，完成题2的工作要求。**

* **实验提交：**

**将完整的源代码和测试截图 粘贴在下面。 测试可使用样例测试。**

* + **代码黏贴处：**

#include <iostream>

#include <string>

#include <iomanip>

#include <cmath>

#include <algorithm>

using namespace std;

int newnum(int n){

    int sum = 0;

    for(int i = 0;i < 3;i++){

        sum = sum \* 10 + n%10;

        n/=10;

    }

    return sum;

}

int main(){

    int n;

    cin >> n;

    while(n--)

    {

        int m;

        cin >> m;

        int \* a = new int [m];

        for(int i = 0;i < m;i++) {

            cin >> a[i];

        }

        for(int i = 1;i < m;i++){

            for(int j = 0;j < m - i;j++){

                if(newnum(a[j]) > newnum(a[j + 1]) ){

                    swap(a[j],a[j+1]);

                }

            }

        }

        for(int i = 0;i < m;i ++ ){

            cout << a[i] << ' ';

        }

        cout <<endl;

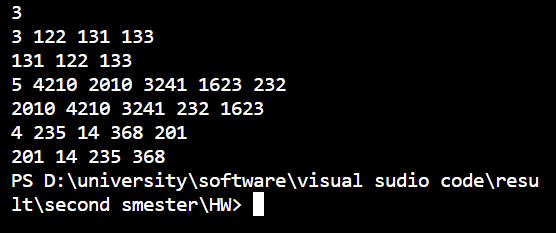
        delete [] a;

    }

    return 0;

}

* + **截图黏贴处：**

****

**3、数组+函数 提高复习部分：（❶❷❸30+❹20分）**

**有如下主函数框架：**

**int main()**

**{ const int n=2;**

**const int k=3**

**const int m=4;**

**int a[n][k], b[k][m];**

**//请补充c的定义**

**InitMatrix(a,n,k); //初始化矩阵a**

**InitMatrix(b,k,m); //初始化矩阵b**

**c=Multiply(a,b,n,k,k,m); //计算aXb ，实际设计可以省略一个参数k**

**Display(c); //输出计算结果, 有可能还需要其他参数，根据自己定义的c调整。**

**return 0;**

**}**

* **实验要求：**

**❶根据主函数的框架理解程序的工作目的，请你根据调用需求设计对应函数的原型并将他们进行实现；**

**❷将你设计的所有函数与主函数进行装配测试，并补充完全所有程序运行需要的内容。**

**❸给出测试的截图。可以使用以下测试数据，也可以自行设计。**

**a (1,2,3; 4,5,6)**

**b(1,2,3,4;2,3,4,5;3,4,5,6)**

**❹思考：若a,b的规模可以由用户设定，那么你的设计需要做哪些调整？给出调整后的源代码和测试截图。（20分）**

* **实验提交：**
  + **代码黏贴处：**

#include <iostream>

#include <string>

#include <iomanip>

#include <cmath>

#include <algorithm>

using namespace std;

void InitMatrix(int \*p,int n,int k){

    for(int i = 0;i < n;i++){

        for(int j = 0;j < k;j++){

            cin >> p[i\*k+j];

        }

    }

}

int\* Multiply(int \* a,int \*b,int n,int k,int m){

    int \* p =new int [n\*m];

    for(int i = 0;i <n;i++){

        for(int j = 0;j < m;j++){

            int sum = 0;

            for(int o = 0;o < k;o++){

                sum += a[i \* k + o]\*b[o \* m +j];

            }

            p[i\*m + j] = sum;

        }

    }

    return p;

}

void Display(int \* p,int n,int m){

    for(int i = 0;i < n;i++){

        for(int j = 0;j < m;j++){

            cout << p[i \* m + j] << ' ';

        }

        cout << endl;

}

}

int main()

{

    const int n=2;

    const int k=3;

    const int m=4;

    int a[n][k], b[k][m];

    int \*c;

    InitMatrix(&a[0][0],n,k);

    InitMatrix(&b[0][0],k,m);

    c=Multiply(&a[0][0],&b[0][0],n,k,m);

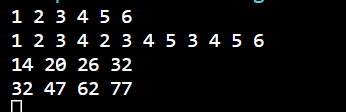
Display(c,n,m);

Delete [] c;

    return 0;

}

* + **截图黏贴处：**

****

**思考❹的**

* **代码黏贴处：**

#include <iostream>

#include <string>

#include <iomanip>

#include <cmath>

#include <algorithm>

using namespace std;

void InitMatrix(int \*p,int n,int k){

    for(int i = 0;i < n;i++){

        for(int j = 0;j < k;j++){

            cin >> p[i\*k+j];

        }

    }

}

int\* Multiply(int \*a,int \* b,int n,int k,int m){

    int \* p = new int [n\*m];

    for(int i = 0;i <n;i++){

        for(int j = 0;j < m;j++){

            int sum = 0;

            for(int o = 0;o < k;o++){

                sum += a[i \* k + o]\*b[o \* m +j];

            }

            p[i\*m + j] = sum;

        }

    }

    return p;

}

void Display(int \* p,int n,int m){

    for(int i = 0;i < n;i++){

        for(int j = 0;j < m;j++){

            cout << p[i \* m + j] << ' ';

        }

        cout << endl;

    }

}

int main()

{

    int n,k,m;

    cout << "依次输入n,k,m:";

    cin >> n >> k >>m;

    int \*a = new int [n\*k];

    int \*b = new int [k\*m];

    int \*c = new int [n\*m];// 就是用一维数组取代二维数组

    InitMatrix(a,n,k);

    InitMatrix(b,k,m);

    c=Multiply(a,b,n,k,m);

    Display(c,n,m);

    delete [] a;

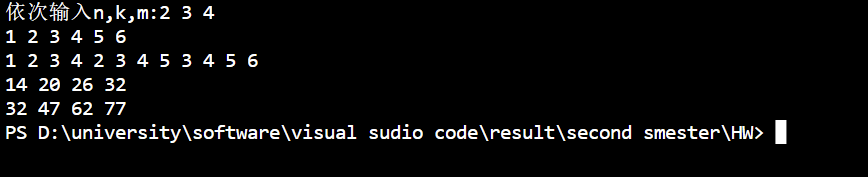
    delete [] b;

    delete [] c;

    return 0;

}

* **截图黏贴处：**

****