C++实验报告

实验一：Rolling Dice

实验目的

1.掌握一维数组的基本用法

2.掌握随机数的使用

3.学习格式化输出的正确使用

实验内容

> Write a program that simulates the rolling of two dice. The program should call rand to roll the first die, and should call rand again to roll the second die. The sum of the two values should then be calculated. [Note: Each die has an integer value from 1 to 6, so the sum of the two values will vary from 2 to 12, with 7 being the most frequent sum and 2 and 12 being the least frequent sums.] Figure 1 shows the 36 possible combinations of the two dice. Your program should roll the two dice 36,000 times. Use a one-dimensional array to tally the numbers of times each sum appears. Print the results in a tabular format. Also, determine if the totals are reasonable

实验方案

```
//实验一：Rolling dice
//步骤一：生成随机数，桶计数
//步骤二：按照要求进行输出
#include<iostream>
#include<iomanip>
#include<cstdlib>
#include<ctime>
using namespace std;
#define rep(i,a,n) for(int i=a;i<=n;i++)
#define frep(i,a,n) for(int i=a;i>=n;i--)
double expected[13]=
{0,0,2.778,5.556,8.333,11.111,13.889,16.667,13.889,11.111,8.333,5.556,2.778};
int sum[13];
int main()
{
    const int ROLLs=36000;
    const int size=12;
    int x,y;
srand(time(0));
rep(i,1,36000)
```

```
{
    int x=rand()%6+1;
    int y=rand()%6+1;
    sum[x+y]++;
}
cout<<setw(10)<<"Sum"<<setw(10)<<"Total"<<setw(10)<<"Expected"<<setw(10)
<<"Atual\n"<<fixed<<showpoint;
rep(i,2,size)
{
    cout<<setw(10)<<i<<setw(10)<<sum[i]<<setprecision(3)<<setw(9)<<expected[i]
<<"%"<<setprecision(3)<<setw(9)<<100.0*sum[i]/36000<<"%\n";
}
    return 0;
}
```

实验结果

```
   Sum     Total  Expected    Atual
     2       980    2.778%    2.722%
     3      2062    5.556%    5.728%
     4      3016    8.333%    8.378%
     5      3947   11.111%   10.964%
     6      5045   13.889%   14.014%
     7      6015   16.667%   16.708%
     8      5034   13.889%   13.983%
     9      3894   11.111%   10.817%
    10      2952    8.333%    8.200%
    11      2072    5.556%    5.756%
    12       983    2.778%    2.731%
```

实验心得

经过该实验，我对C++的格式化输出的具体应用有了更深入的了解，同时也掌握了随机数种子的使用方法，并且对数组的初始化也有了进一步的理解，如果数组放置在全局，则会初始化为01的特点，是一个初学者常常会犯错的地方，需要引起注意。

实验二：Array（数组）

实验目的

1.掌握函数声明、定义和调用的方法

2.学习灵活使用数组。

3.分支结构的使用

实验内容

Task 1:

> Create an array with N integers. Input the values of each element. Then output the size of the array and values of each element.

Task 2:

> Modify the program to use function *input* to input the value to an array, and function *output* to output an array. Sample input and output are same with Task 1.

Task 3:

> Add functions to the program.
>
> 1. Function *search* finds the position of “key value” in the array. If not found, return -1.
> 2. Function *minimum* gives the minimum value of the array.
> 3. Function *maximum* gives the maximum value of the array.
> 4. Function minipos gives the position of minimum value in the array.
> 5. Function *maxipos* gives the position of maximum value in the array.
>
> Write some codes to test your functions. Maybe you can try replacing keyboard input with random numbers

Task 4:

> Add functions to the program.
>
> 1. Function *sum* sums up the values of every elements.
> 2. Function *average* calculates the average value of elements, calling function *sum* in the code.
> 3. Function *even* counts the number of even numbers.
> 4. Function *odd* counts the number of odd numbers.
>
> Write some codes to test your functions.

Task 5:

> Add functions to the program.
>
> 1. Function *split* divides the array into two arrays: odd array and even array.
> 2. Function *combine* merges two arrays into one array, one array following another.
> 3. (*)Function *combineordered* merges two ordered arrays into one array, The merged array is still in order.

实验方案

```
//实验二：
#include<iostream>
#include<algorithm>
using namespace std;
const int N = 1e6;
int a[N]; //主要数组，也就是用户一直操作的数组
int even[1000000];
int odd[1000000];
int c[N];
int n;
#define rep(i,a,n) for(int i=a;i<=n;i++)
```

```
#define frep(i,a,n) for(int i=a;i>=n;i--)
//任务二
void input()
{
    cin >> n;
    rep(i, 1, n)
    {
        cin >> a[i];
    }
    rep(i, 1, n)
    {
        cout << a[i] << ' ';
    }
    cout << '\n';
}
//任务三
int find(int x, int a[], int n)//x表示要查找的值
{
    rep(i, 1, n)
    {
        if (a[i] == x)
        {
            return i;
        }
    }
    return -1;
    //没有找到请返回-1
}
//返回最小值
int miniumum(int a[], int n)
{
    int minn = 999999999;
    rep(i, 1, n)
    {
        if (a[i] < minn)
        {
            minn = a[i];
        }
    }
    return minn;
}
//返回最大值
int maximum(int a[], int n)
{

    int maxn = -999999999;
    rep(i, 1, n)
    {
        if (a[i] > maxn)
        {
            maxn = a[i];
        }
    }
    return maxn;
}
//返回最小值所在
```

```
int minipos(int a[], int n)
{
    int minn = 999999999;
    int minpos;
    rep(i, 1, n)
    {
        if (a[i] < minn)
        {
            minn = a[i];
            minpos = i;

        }
    }
    return minpos;
}
//返回最大值
int maxipos(int a[], int n)
{
    int maxn = -999999999;
    int maxpos;
    rep(i, 1, n)
    {
        if (a[i] > maxn)
        {
            maxn = a[i];
            maxpos = i;
        }
    }
    return maxpos;
}
//任务四：
//求和函数
int sum(int a[], int n)
{
    int ans = 0;
    rep(i, 1, n)
    {
        ans += a[i];
    }
    return ans;
}
//平均值函数，调用sum函数
int average(int a[], int n)
{
    return sum(a, n) * 1.0 / n;
}
//计算奇数
int even_(int a[], int n)
{
    int cnt = 0;
    rep(i, 1, n)
    {
        if (a[i] % 2 == 0)
        {
            cnt++;
        }
```

```
    }
    return cnt;
}
//计算偶数
int odd_(int a[], int n)
{
    return n - even_(a, n);
}
//任务五：
//拆分函数
void chaifen(int a[], int n)
{
    //一个是传入的数组。一个则是数组的大小
       //默认从一开始
    int cnt_even = 0;
    int cnt_odd = 0;
    rep(i, 1, n)
    {
        if (a[i] % 2 == 0)
        {
            even[++cnt_even] = a[i];
        }
        else {
            odd[++cnt_odd] = a[i];
        }
    }
    cout << cnt_odd << '\n';
    rep(i, 1, cnt_odd)
    {
        cout << odd[i] << ' ';
    }
    cout << '\n';
    cout << cnt_even << '\n';
    rep(i, 1, cnt_even)
    {
        cout << even[i] << ' ';
    }
    cout << '\n';
}
//合并
void combine(int a[], int a_cnt, int b[], int b_cnt)
{
    int cnt = 0;
    rep(i, 1, a_cnt)
    {
        c[++cnt] = a[i];
    }
    rep(i, 1, b_cnt)
    {
        c[++cnt] = b[i];
    }
    cout << "合并后的数组是" << '\n';
    rep(i, 1, a_cnt + b_cnt)
    {
        cout << c[i] << ' ';
    }
```

```
    cout << '\n';
    //接长数组
}
//合并有序数组
void ssort(int a[], int a_cnt, int b[], int b_cnt)
{
    int cnt = 0;
    rep(i, 1, a_cnt)
    {
        c[++cnt] = a[i];
    }
    rep(i, 1, b_cnt)
    {
        c[++cnt] = b[i];
    }
    sort(c + 1, c + 1 + a_cnt + b_cnt);
    cout << "合并后的数组是" << '\n';
    rep(i, 1, a_cnt + b_cnt)
    {
        cout << c[i] << ' ';
    }
    cout << '\n';

}
//合并有序数组
void nsort(int a[], int a_cnt, int b[], int b_cnt)
{
    int cnt = 0;
    rep(i, 1, a_cnt)
    {
        c[++cnt] = a[i];
    }
    rep(i, 1, b_cnt)
    {
        c[++cnt] = b[i];
    }
    sort(c + 1, c + 1 + a_cnt + b_cnt);
    cout << "合并后的数组是" << '\n';
    frep(i, a_cnt + b_cnt, 1)
    {
        cout << c[i] << ' ';
    }
    cout << '\n';

}
//判断素数
bool isprime(int num)
{
    if (num == 1)
    {
        return 0;
    }
    if (num == 2)
    {
        return 1;
    }
```

```
    if (num % 2 == 0)
    {
        return 0;
    }
    for (int i = 2; i * i <= num; i++)
    {
        if (num % i == 0)
        {
            return 0;
        }

    }
    return 1;
}
int main()
{
    /*
        任务一:
         int n;
         cin>>n;
         rep(i,1,n)
         {
            cin>>a[i];
         }
        rep(i,1,n)
        {
            cout<<a[i]<<' ';
        }
    */
    //测试过程
    cout << "接下来开始实验二" << '\n';
    cout << '\n';
    cout << "请输入数组元素个数" << '\n';
    cout << "\n";
    cout << "请再输入元素" << '\n';
    input();
    cout << "请输入你要查找的数字，sir" << '\n';
    cout << '\n';
    int x_;
    cin >> x_;
    int x = find(x_, a, n);
    if (x == -1)
    {
        cout << "找不到" << '\n';
        cout << '\n';
    }
    else
    {
        cout << "找到了,是" << x << '\n';
        cout << '\n';
    }
    int minn = miniumum(a, n);
    int maxn = maximum(a, n);
    cout << "接下来要查找最小值还是最大值呢，少侠？" << '\n';
    cout << '\n';
    cout << "1表示最小值" << '\n';
```

```
    cout << '\n';
    cout << "2表示最大值" << '\n';
    cout << '\n';
    cout << "3表示2个都输出哦" << '\n';
    cout << '\n';
    int cnt_mm;
    cin >> cnt_mm;
    if (cnt_mm == 1)
    {
        cout << "最小值是" << minn << '\n';
        cout << '\n';
    }
    else if (cnt_mm == 2)
    {
        cout << "最大值是" << maxn << '\n';
        cout << '\n';
    }
    else
    {
        cout << "最小值是" << minn << " " << "最大值是" << maxn << '\n';
        cout << '\n';
    }
    cout << "接下来要查找最小值的下标还是最大值的下标呢" << '\n';
    cout << '\n';
    cout << "1表示最小值的" << '\n';
    cout << '\n';
    cout << "2表示最大值的" << '\n';
    cout << '\n';
    cout << "3表示2个都输出" << '\n';
    cout << '\n';
    int maxpos = maxipos(a, n);
    int minpos = minipos(a, n);
    int cnt_pos;
    cin >> cnt_pos;
    if (cnt_pos == 1)
    {
        cout << "最小值的下标是" << minpos << '\n';
        cout << '\n';
    }
    else if (cnt_pos == 2)
    {
        cout << "最大值的下标是" << maxpos << '\n';
        cout << '\n';
    }
    else
    {
        cout << "最大值的下标是" << maxpos << ' ' << "最小值的下标是" << minpos <<
'\n';
        cout << '\n';
    }
    cout << "少侠要求和吗" << '\n';
    cout << '\n';
    cout << "求和请输入1" << '\n';
    cout << '\n';
    cout << "不求和输入2" << '\n';
    cout << '\n';
```

```
    int cnt_sum;
    cin >> cnt_sum;
    if (cnt_sum == 1)
    {
        int ans = sum(a, n);
        cout << "数组的和是" << ans << '\n';
        cout << '\n';
    }
    cout << '\n';
    cout << "少侠要求平均吗" << '\n';
    cout << '\n';
    cout << "求平均请输入1" << '\n';
    cout << '\n';
    cout << "不平均请输入2" << '\n';
    cout << '\n';
    int cnt_aver;
    cin >> cnt_aver;
    if (cnt_aver==1)
    {
        int ave = average(a, n);
        cout << "数组的平均值是" << ave << '\n';
        cout << '\n';
    }
    int cnt_e = even_(a, n);
    int cnt_o = odd_(a, n);
    cout << "输入一个数如果是奇数，我就输出奇数，否则输入偶数,如果这个数还是质数，那两个都输出"
<< '\n';
    cout << '\n';
    int cnt_xx;
    cin >> cnt_xx;
    if (isprime(cnt_xx))
    {
        cout << "奇数的个数是" << cnt_e << "偶数的个数是" << cnt_o << '\n';
        cout << '\n';
    }
    else if (cnt_xx & 1)
    {
        cout << "奇数的个数是" << cnt_e << '\n';
        cout << '\n';
    }
    else {
        cout << "偶数的个数是" << cnt_o << '\n';
        cout << '\n';
    }
    cout << "请输入你要合并的数组" << '\n';
    cout << '\n';
    cout << "先输入元素个数，再输入数组" << '\n';
    cout << '\n';
    int cnt_ceshi;
    cin >> cnt_ceshi;
    int ceshi[10000];
    rep(i, 1, cnt_ceshi)
    {
        cin >> ceshi[i];
    }
    cout << "请问是要无序合并还是顺序合并还是说逆序合并呢" << '\n';
```

```
    cout << '\n';
    cout << "1表示无序" << '\n';
    cout << '\n';
    cout << "2表示顺序" << '\n';
    cout << '\n';
    cout << "3表示逆序" << '\n';
    cout << '\n';
    int cnt_sort;
    cin >> cnt_sort;
    if (cnt_sort == 1)
    {
        combine(a, n, ceshi, cnt_ceshi);
    }
    else if (cnt_sort == 2)
    {
        ssort(a, n, ceshi, cnt_ceshi);
    }
    else
    {
        nsort(a, n, ceshi, cnt_ceshi);
    }
    cout << "世间温柔\n";
    cout << '\n';
    cout << "不过是芳春柳摇染花香" << '\n';
    cout << '\n';
    cout << "槐序蝉鸣入深巷" << '\n';
    cout << '\n';
    cout << "白茂叶落醉故乡" << '\n';
    cout << '\n';
    cout << "爱笑的龙猫出品\n";
    return 0;
}
```

实验结果

> 接下来开始实验二
>
> 请输入数组元素个数
>
> 请再输入元素
> 5
> 1 2 3 4 5
> 1 2 3 4 5
> 请输入你要查找的数字，sir
>
> 2
> 找到了,是2
>
> 接下来要查找最小值还是最大值呢，少侠?
>
> 1表示最小值
>
> 2表示最大值
>
> 3表示2个都输出哦

3
最小值是1 最大值是5

接下来要查找最小值的下标还是最大值的下标呢

1表示最小值的

2表示最大值的

3表示2个都输出

3
最大值的下标是5 最小值的下标是1

少侠要求和吗

求和请输入1

不求和输入2

1
数组的和是15

少侠要求平均吗

求平均请输入1

不平均请输入2

1
数组的平均值是3

输入一个数如果是奇数，我就输出奇数，否则输入偶数,如果这个数还是质数，那两个都输出

3
奇数的个数是2偶数的个数是3

请输入你要合并的数组

先输入元素个数，再输入数组

5
1 2 3 4 5
请问是要无序合并还是顺序合并还是说逆序合并呢

1表示无序

2表示顺序

3表示逆序

3
合并后的数组是
5 5 4 4 3 3 2 2 1 1
世间温柔

不过是芳春柳摇染花香

槐序蝉鸣入深巷

白茂叶落醉故乡

爱笑的龙猫出品

实验心得

> 通过实验二，我掌握了数组传入函数的方法，对全局变量和局部变量的作用域范围有了更好的理解。除此之外，一开始我并没有考虑到提示用户的代码，于是，在实验二，我更加理解用户的需求，实际上这也正是软件工程的目标之一，要以用户为主体进行设计，也正是我所追求的目标之一。

实验设计总结

> 1.完成本实验体现了程序员必须具备的细心和耐心的品质，想要完成一项出色的项目，必须做到精益求精，反复操作与实践。
>
> 2.在经过本次实验课，我对C++的了解更加深刻，很多知识得到了巩固，一些不懂的知识也渐渐明了，这其中对数组和函数的收获很多，尤其是数组的各种细节，都有所发现。

实验日期

> 11月10日