

### 实验三

1.

```
#include <iostream>
```

```
using namespace std;
```

```
void fun()
```

```
{
```

```
    int n, count=0;
```

```
    cout<<"输入 n: ";
```

```
    cin>>n;
```

```
    for (int i = 0; i <= n/5; i++)
```

```
    {
```

```
        for (int j = 0; j <= (n-(5*i))/2; j++)
```

```
        {
```

```
            count++;
```

```
            int k = n - (5*i) - (2*j);
```

```
            cout<<i<<" * 5 + "<<j<<" * 2 + "<<k<<" = "<<n<<endl;
```

```
        }
```

```
    }
```

```
    cout<<"一共有 "<<count<<" 种兑换方法"<<endl;
```

```
}
```

```
int main(int argc, char const *argv[])
```

```
{
```

```
    fun();
```

```
    return 0;
```

```
}
```

运行结果：

```
fisher — question1 — 85x24
Last login: Sun Mar 31 20:33:55 on ttys000
/Volumes/HD/OneDrive\ -\ m.scnu.edu.cn/算法设计与分析/实验三/question1 ; exit;
FisherdeMBP:~ fisher$ /Volumes/HD/OneDrive\ -\ m.scnu.edu.cn/算法设计与分析/实验三/qu
estion1 ; exit;
输入 n: 10
0 * 5 + 0 * 2 + 10 = 10
0 * 5 + 1 * 2 + 8 = 10
0 * 5 + 2 * 2 + 6 = 10
0 * 5 + 3 * 2 + 4 = 10
0 * 5 + 4 * 2 + 2 = 10
0 * 5 + 5 * 2 + 0 = 10
1 * 5 + 0 * 2 + 5 = 10
1 * 5 + 1 * 2 + 3 = 10
1 * 5 + 2 * 2 + 1 = 10
2 * 5 + 0 * 2 + 0 = 10
一共有 10 种兑换方法
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
█
```

2.

```
# include <iostream>
```

```
# include <vector>
```

```
using namespace std;
```

```
void fun2()
```

```
{
```

```
    int minSize, maxSize, n, fishSize[51], min, max;
```

```
    cout<<"fish's minSize and maxSize: ";
```

```
    cin>>minSize>>maxSize;
```

```
    cout<<"fish number n: ";
```

```
    cin>>n;
```

```
    cout<<"fish's size: ";
```

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

```
        cin>>fishSize[i];
```

```
    }
```

```

    if (n>0) {
        min = fishSize[0]*2;
        max = fishSize[0]*10;
    }

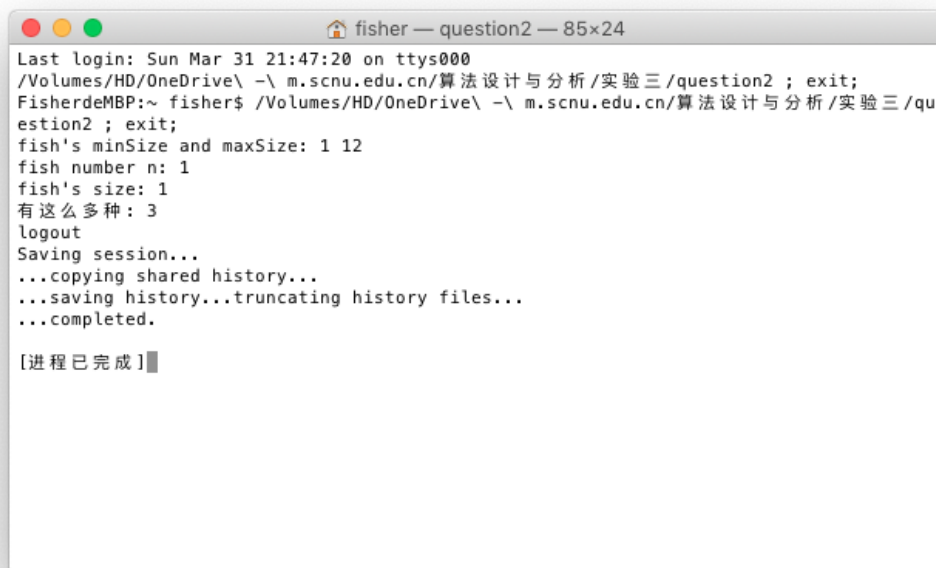
    for (int i = 1; i < n; i++) {
        if (fishSize[i]*2 < min) {
            min = fishSize[i]*2;
        }
        if (fishSize[i]*10 > max) {
            max = fishSize[i]*10;
        }
    }

    cout<<"有这么多种: "<<maxSize-minSize-(max-min)<<endl;
}

int main(int argc, char const *argv[])
{
    fun2();
    return 0;
}

```

运行结果:

A screenshot of a macOS terminal window titled "fisher — question2 — 85x24". The window shows the execution of a program named "fisher". The output includes login information, file paths, and program-specific data like "fish's minSize and maxSize: 1 12", "fish number n: 1", and "fish's size: 1". It also shows session saving progress and ends with "[进程已完成]" (Process completed).

```
fisher — question2 — 85x24
Last login: Sun Mar 31 21:47:20 on ttys000
/Volumes/HD/OneDrive\ -\ m.scnu.edu.cn/算法设计与分析/实验三/question2 ; exit;
FisherdeMBP:~ fisher$ /Volumes/HD/OneDrive\ -\ m.scnu.edu.cn/算法设计与分析/实验三/qu
estion2 ; exit;
fish's minSize and maxSize: 1 12
fish number n: 1
fish's size: 1
有这么多种: 3
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[进程已完成]
```

3.

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

```
void fun3()
{
    int n, a[1001], maxArea=0;

    cout<<"输入 n: ";
    cin>>n;
    cout<<"输入高度: ";
    for (int i = 0; i < n; i++) {
        cin>>a[i];
    }

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

```

int area = a[i];

// 向前找矩形，直到找到的矩形比当前矩形小
for (int j = i-1; j >= 0; j--)
{
    if (a[j] >= a[i]) {
        area += a[i];
    } else {
        break;
    }
}

// 向后找矩形，直到找到的矩形比当前矩形小
for (int j = i+1; j < n; j++)
{
    if (a[j] >= a[i]) {
        area += a[i];
    } else {
        break;
    }
}

if (area > maxArea)
{
    maxArea = area;
}
}


cout<<"maxArea: "<<maxArea<<endl;
}

int main(int argc, char const *argv[])
{

```

```
fun3();  
return 0;  
}
```

运行结果：



```
fisher — question3 — 85x24  
Last login: Sun Mar 31 22:31:12 on ttys000  
/Volumes/HD/OneDrive\ -\ m.scnu.edu.cn/算法设计与分析/实验三/question3 ; exit;  
FisherdeMBP:~ fisher$ /Volumes/HD/OneDrive\ -\ m.scnu.edu.cn/算法设计与分析/实验三/qu  
estion3 ; exit;  
输入n: 6  
输入高度: 3 1 6 5 2 3  
maxArea: 10  
logout  
Saving session...  
...copying shared history...  
...saving history...truncating history files...  
...completed.  
  
[进程已完成]
```