

# 实验一

## 实验名称：蛮力法及分治法

### 实验要求：

实验内容包含以下题目。

要求提交一份实验报告，对每个题目内容包括：**思路描述，代码，运行截图**  
(下一页附实验报告模板)

### 题目 1：0/1 背包问题

实现课本 P44 页的 0/1 背包问题。

**输入格式：** 给定数字  $n$  代表物品数量以及  $C$  代表背包容量，接着一行给定  $n$  个物品的重量，接着一行给定  $n$  个物品的价格。

**输出格式：** 输出一个数字，代表所能拿取的物品的最大价值。

**输入样例：** 4 10  
7 3 4 5  
42 12 40 25

**输出样例：** 65

### 题目 2：数字游戏

把数字 1, 2, 3..., 9 填入下面含有四则运算的算式中，每个数字只出现一次，且数字 1 不出现在乘法和除法的“一位数”那个格中。输出所有可能答案。

提示：使用蛮力法，生成排列

$$\square\square\times\square+\square\square\square\div\square-\square\square=0$$

### 题目 3：折半查找区间

给定一个数组（所有数据升序排列给出），并给定一个闭区间  $[a, b]$

使用减治法的思想，输出  $a$  到  $b$  之间（包含  $a$  和  $b$ ，闭区间）的所有数值

**输入格式：** 给定一个数字  $n$ ，接着一行给出  $n$  个数字（已经处于升序排列），接着给出  $a$  和  $b$  的值。

**输出格式：** 输出所有  $[a, b]$  闭区间范围内的数值

**输入样例：** 10  
2 5 8 13 17 25 35 36 47 50  
16 36

**输出样例：** 17 25 35 36

# 实验报告

课程名称	算法设计与分析	班级	21 杨班
实验编号	一	姓名	皮昊旋
实验名称	实验一	学号	2021428010127

## 题目 1： 题目名称...

### ➤ 思路描述

- 1、按照格式输入数据
- 2、创建动态规划表并置零
- 3、检查动态规划表，如果放入当前货物使得总值最大，则放入，否则不放入

### ➤ 代码

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    int length,k,k1;
    scanf("%d",&length);
    int a[length];
    for(int i=0;i<length;i++) scanf("%d",&a[i]);
    int m,n;
    scanf("%d%d",&m,&n);
    int low=0,high=length-1;
    int mid;
    while(low<high){
        mid=(low+high)/2;
        if(m<a[mid]&&m<a[mid-1]) high=mid-1;
        if(m>a[mid]&&m>a[mid+1]) low=mid+1;
        if(m>a[mid]&&m<=a[mid+1]){
            k=mid+1;
            break;}
        if(m<=a[mid]&&m>a[mid-1]){
            k=mid;
            break;}
    }
}
```

```

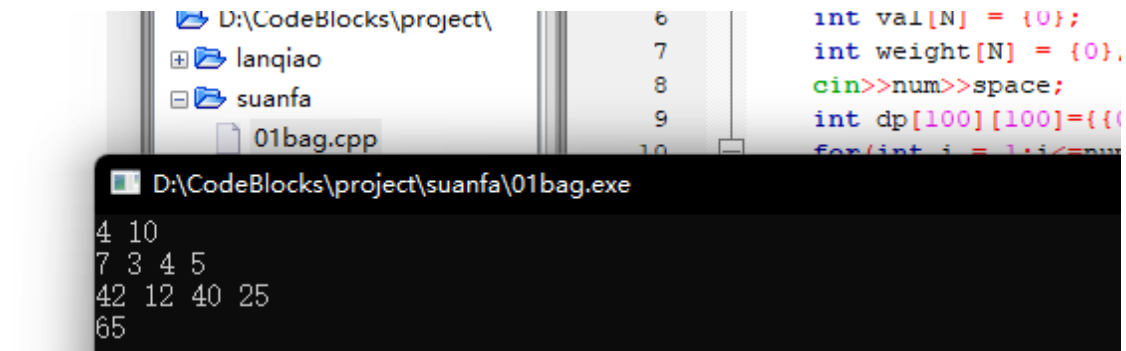
int low1=0,high1=length-1;
int mid1;
while(low1<high1){
    mid1=(low1+high1)/2;
    if(n<a[mid1]&& n<a[mid1-1]) high1=mid1-1;
    if(n>a[mid1]&& n>a[mid1+1]) low1=mid1+1;
    if(n>a[mid1]&& n<=a[mid1+1]){
        k1=mid1+1;
        break;}
    if(n<=a[mid1]&& n>a[mid1-1]){
        k1=mid1;
        break;}
}
}
for(int i=k;i<=k1;i++) printf("%d ",a[i]);

return 0;

}

```

### ➤ 运行截图



## 题目 2: 题目名称...

### ➤ 思路描述

直接 for 循环，循环所有数字，都从 0 到 9，没一个循环添加限制条件，不与之前的数字相等，然后检验结果，如果结果为 0，则输出

### ➤ 代码

```

#include <bits/stdc++.h>
using namespace std;
int main(){
    float a = 0,b = 0,c = 0,d = 0,e = 0,f = 0,g = 0,h = 0,i = 0,temp = 0;
    float res = (10*a+b)*c+(100*d+10*e+f)/(1.0*g)-(10*h+i);
    for(a=1;a<=9;a++){

```

```

//cout<<"a:"<<a<<endl;
//cin>>temp;
for(b=1;b<=9;b++){
    if(b!=a){
        //cout<<"b:"<<b<<endl;
        //cin>>temp;
        for(c=2;c<=9;c++){
            if(c!=a && c!=b){
                //cout<<"c:"<<c<<endl;
                //cin>>temp;
                for(d = 1;d<=9;d++){
                    if(d!=c&&d!=b&&d!=a){
                        //cout<<"d:"<<d<<endl;
                        //cin>>temp;
                        for(e=1;e<=9;e++){
                            if(e!=d&&e!=c&&e!=b&&e!=a){
                                //cout<<"E:"<<e<<endl;
                                //cin>>temp;
                                for(f=1;f<=9;f++){

if(f!=a&&f!=b&&f!=c&&f!=d&&f!=e){

//cout<<"F:"<<f<<endl;
//cin>>temp;
for(g=2;g<=9;g++){

if(g!=a&&g!=b&&g!=c&&g!=d&&g!=e&&g!=f){

//cout<<"G:"<<g<<endl;

//cin>>temp;

for(h=1;h<=9;h++){

if(h!=a&&h!=b&&h!=c&&h!=d&&h!=e&&h!=f&&h!=g){

//cout<<"H:"<<h<<endl;

//cin>>temp;

for(i=1;i<=9;i++){

if(i!=a&&i!=b&&i!=c&&i!=d&&i!=e&&i!=f&&i!=g&&i!=h){

//cout<<"I:"<<i<<endl;

```

```

res = (10*a+b)*c+(100*d+10*e+f)/(1.0*g)-(10*h+i);

//cout<<res<<endl;

if(res ==0){

cout<<a<<b<<"¡Á"<<c<<"+"<<d<<e<<f<<"/"<<g<<"-"<<h<<i<<endl;

}

}

}

//if(res

==0)break;

}

//if(res ==0)break;

}

//if(res ==0)break;

}

//if(res ==0)break;

}

//if(res ==0)break;

}

//if(res ==0) break;

}

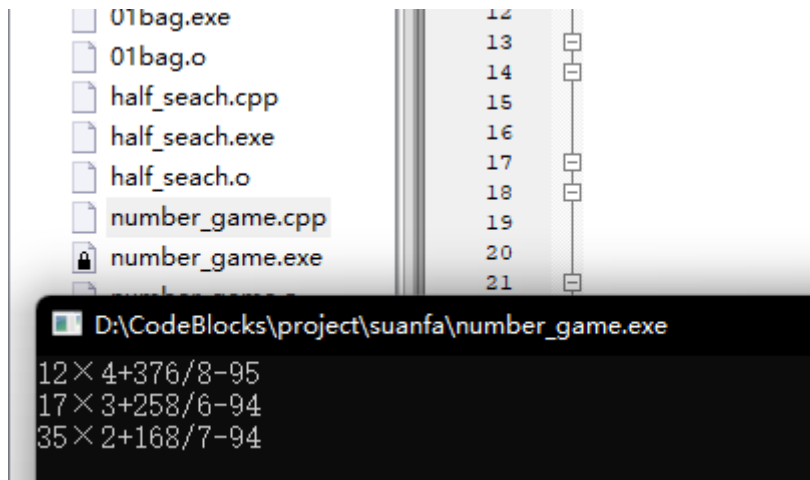
//if(res ==0) break;

}

return 0;
}

```

➤ 运行截图



### 题目 3: 题目名称...

➤ 思路描述

- 1、输入数据
- 2、While 循环寻找两个最接近目标值的数的下标
- 3、循环输出两个目标值下标之间的数字

➤ 代码

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main(){
```

```
    int length,k,k1;
```

```
    cin>>length;
```

```
    int a[length];
```

```
    for(int i=0;i<length;i++) cin>>a[i];
```

```
    int m,n;
```

```
    cin>>m>>n;
```

```
    int low=0,high=length-1;
```

```
    int mid;
```

```
    while(low<high){
```

```
        mid=(low+high)/2;
```

```
        if(m<a[mid]&&m<a[mid-1]) high=mid-1;
```

```
        if(m>a[mid]&&m>a[mid+1]) low=mid+1;
```

```
        if(m>a[mid]&&m<=a[mid+1]){
```

```
            k=mid+1;
```

```
            break;} 
```

```
        if(m<=a[mid]&&m>a[mid-1]){
```

```
            k=mid;
```

```
            break;
```

```
    }
```

```
 }
```

```
int low1=0,high1=length-1;
```

```
int mid1;
```

```

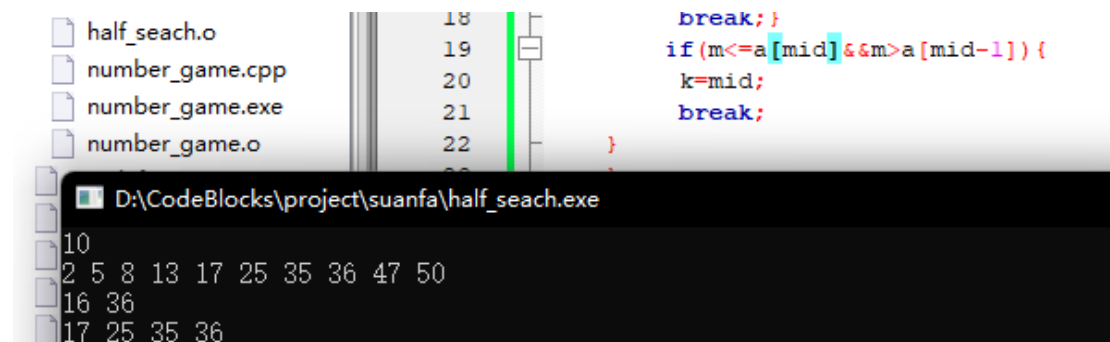
while(low1<high1){
    mid1=(low1+high1)/2;
    if(n<a[mid1]&&n<a[mid1-1]) high1=mid1-1;
    if(n>a[mid1]&&n>a[mid1+1]) low1=mid1+1;
    if(n>a[mid1]&&n<=a[mid1+1]){
        k1=mid1+1;
        break;}
    if(n<=a[mid1]&&n>a[mid1-1]){
        k1=mid1;
        break;}
}
}
for(int i=k;i<=k1;i++) {
    if(i==k1)cout<<a[i];
    else{
        cout<<a[i]<<" ";
    }
}

return 0;

}

```

### ➤ 运行截图



The screenshot shows a code editor with a file explorer on the left containing files: half\_seach.o, number\_game.cpp, number\_game.exe, and number\_game.o. The code editor displays a snippet of C++ code with line numbers 18 to 22. Line 19 shows a break statement, and line 20 shows an if statement: `if (m<=a[mid] && m>a[mid-1]) {`. Below the code editor is a terminal window titled `D:\CodeBlocks\project\suanfa\half_seach.exe`. The terminal output shows the number 10, followed by a space-separated list of numbers: 2 5 8 13 17 25 35 36 47 50. Then, the number 16 is shown, followed by 36. Finally, the numbers 17 25 35 36 are shown.