**大 连 海 事 大 学**

**\_\_\_\_\_\_\_\_\_\_\_\_ 实 验 报 告**

**算法分析与设计**

|  |
| --- |
| 专 业 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 姓名\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 学号\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **电子信息**  **1120211433**  **杨显鹏**  实验日期\_\_\_\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_**\_\_ 指导教师\_\_**\_\_\_\_\_\_\_\_\_\_\_**\_\_\_\_\_\_\_  **2021.11.06**  **曲衍鹏** |
| 第\_\_\_\_\_\_\_\_实验 实验名称 \_\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**\_\_\_\_\_\_\_\_\_  **最大乘积问题**  **一次** |

**实验一：最大乘积问题**

**一、实验目的**

掌握数组的运用，并熟悉基础的代码编写与运行。

**二、实验内容**

(1) 设n是一个正整数，现在要求将n分解为若干个互不相同的自然数的和，并且使这些自然数的乘积最大，对于给定的正整数n，计算最优分解方案。

提示：

1.把一个正整数从中间分开（如果是偶数直接除以2如果是奇数，分别加1除以2，减1除以2）

2.其中一部分保留在A[]数组中（奇数的话，比较大的那一部分保留给A[]数组），另一部分赋给temp，并重复1，2 步骤

3.最后把temp赋给A[]数组

(2) 要求至少构造10组数据，及运行结果。

**三、实验数据**

实验数据随机选择10个整数如表1所示：

表1

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验数据编号 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 实验数据值 | 10 | | 22 | 15 | 14 | 13 | 16 | 17 | 33 | 25 | 20 |

**四、程序代码**

#include <stdio.h>

#define MAX 100

int get(int n)

{

int a[MAX];

int x=2;

int index=0;

a[index++]=x;

n-=x;

//将n分解放进数组

while(n>a[index-1]){

a[index]=a[index-1]+1;

n-=a[index];

index++;

}

int num=index-1;

//将余量加入从后向前分成1加入数组

while(n){

a[num]++;

num=(num-1+index)%(index);

n--;

}

//输出数组

printf("分解出的数组为：");

for(int i=0;i<index;i++){

printf("%d;",a[i]);

}

//计算最大乘积

int result=1;

for(int i=0;i<index;i++){

result\*=a[i];

}

printf("\n\n最大乘积为：%d\n",result);

return 0;

}

int main(){

int a;

int n[10]={10,22,15,14,13,16,17,33,25,20};

printf("10个测试的数\n");

for(int i=0;i<=9;i++){

printf("%d;",n[i]);

}

printf("\n");

for(int i=0;i<=9;i++){

a=n[i];

get(a);

}

return 0;

}

**五、总结与体会**

**5.1实验结果**

最终实验结果如图1所示，采用随机选取的10个整数，计算出其的分解数组与最大乘积。



图1

**5.2分析**

本实验问题比较简单，首先讲要测试的数作为输入传入程序中，将数据按照提示方法一步一步进行分解。并将其乘积计算出来。

通过本实验，对算法有了一定的了解。算法（Algorithm）是指解题方案的准确而完整的描述，是一系列解决问题的清晰指令，算法代表着用系统的方法描述解决问题的策略机制。也就是说，能够对一定规范的输入，在有限时间内获得所要求的输出。如果一个算法有缺陷，或不适合于某个问题，执行这个算法将不会解决这个问题。不同的算法可能用不同的时间、空间或效率来完成同样的任务。一个算法的优劣可以用空间复杂度与时间复杂度来衡量。