深 圳 大 学 实 验 报 告

课程名称：软件工程

实验项目名称：实验四：模块过程设计

学院：计算机与软件学院

专业：软件工程

指导教师：杜文峰

报告人：__学号：__班级：__

实验时间：2025 年 11 月 3 日

教务部制

实验目的与要求：

实验目的:

（1）了解模块过程分析方法

（2）掌握程序流图绘制方法

（3）了解程序流程图绘制工具的使用

实验要求：

（1）分析附件中给出的 C++程序源代码

（2）使用“万兴图示”完成该程序的程序流程图

（3）使用人工智能平台，分析代码内容，生成 Mermaid 绘图代码，并绘制对应流程图

（4）比较绘制流程图与 AI 生成流图之间的区别，并写一段感想。

实验过程及内容：

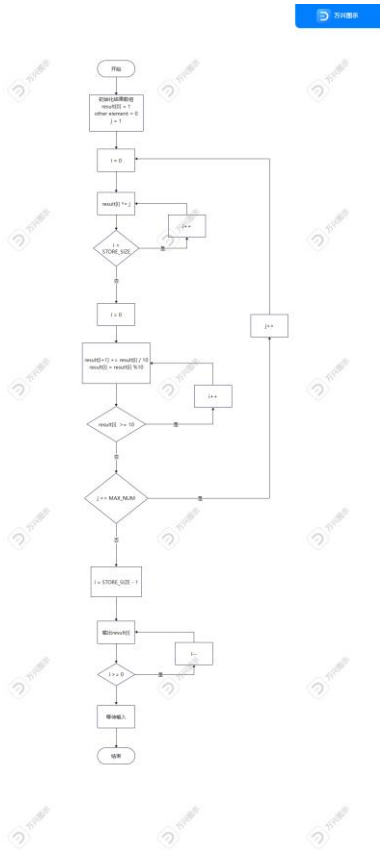
源代码：

```
// jiechen.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"

#define MAX_NUM 1000
#define STORE_SIZE    3000

int main(int argc, char* argv[])
{
    //Initialize the result and Set the last key
    int result[STORE_SIZE] = {1};

    //Begin compute the result
    for(int j = 1; j <= MAX_NUM; j++)
    {
        //Times each key with the new number
        for(i = 0; i < STORE_SIZE; i++)
        {
            result[i] *= j;
        }

        //We will add the
        for(i = 0; i < STORE_SIZE; i++)
        {
            if(result[i] >= 10)
            {
                //Add the result of 10 times to the high key
```

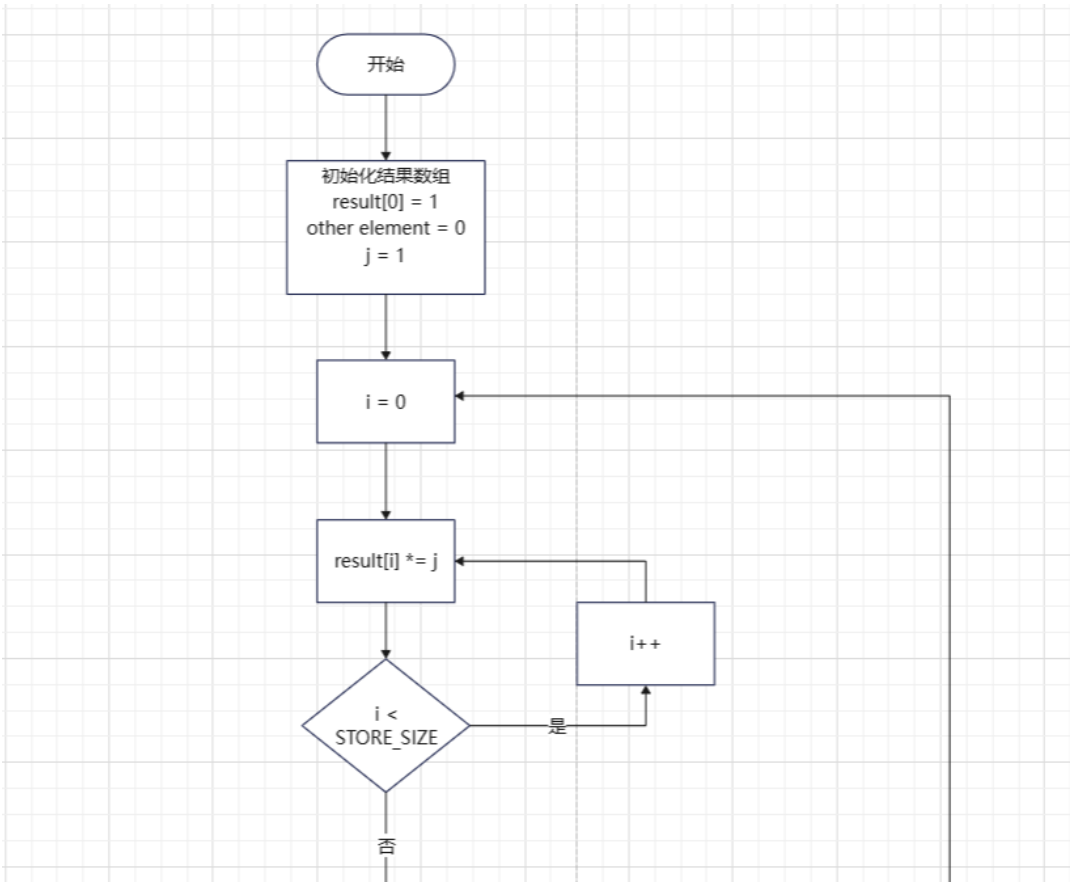
```
                    result[i + 1] += result[i] / 10;

                    //Get the value of current key.
                    result[i] = result[i] % 10;
                }

            }
        }

        //print the result
        for(i = STORE_SIZE - 1; i >= 0; i--)
        {
            printf("%d",result[i]);
        }

        //pause
        getchar();
        return 0;
}
```

可以看出，这段代码实现的功能是计算 1000 的阶乘，并且将计算结果以逆序形式存储在数组 result 中。

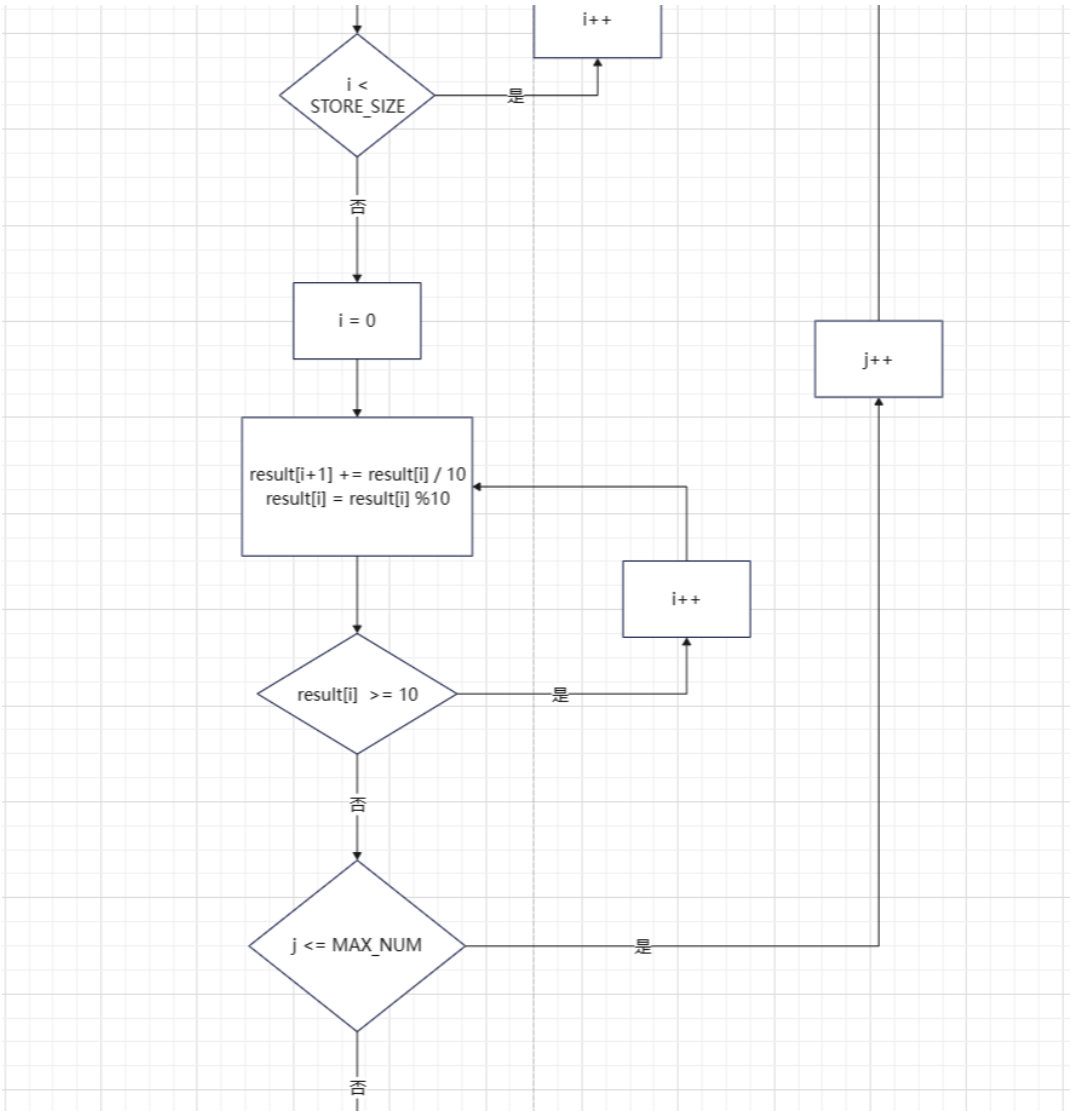
根据代码的逻辑，我们可以绘制出其流程图：



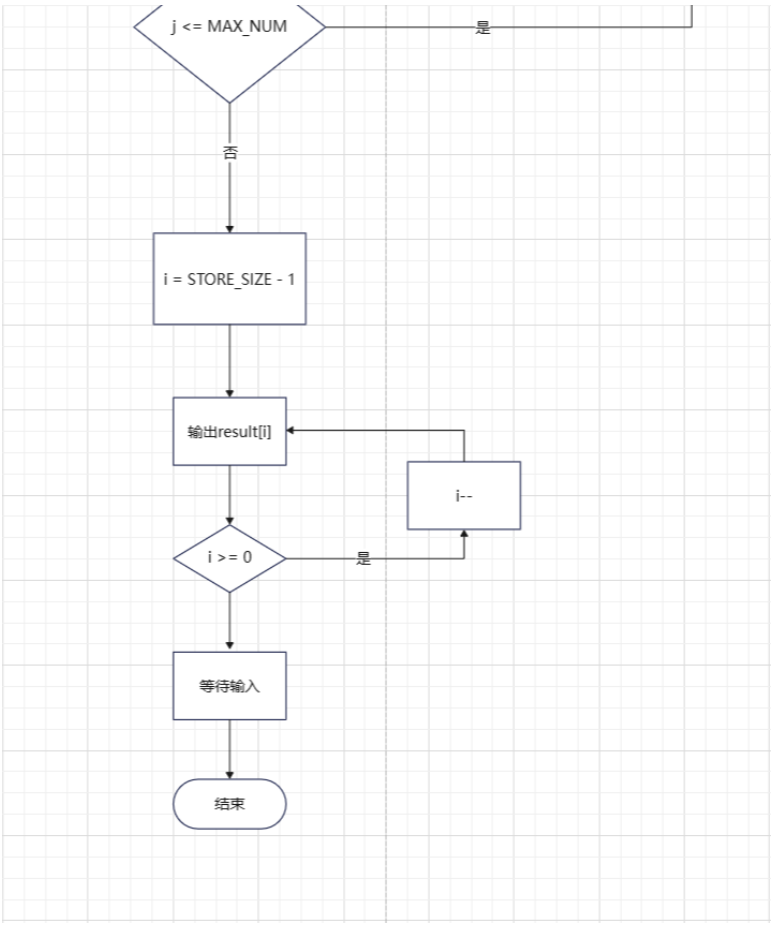
完成流程图可查看附件 1000jiecheng.jpg



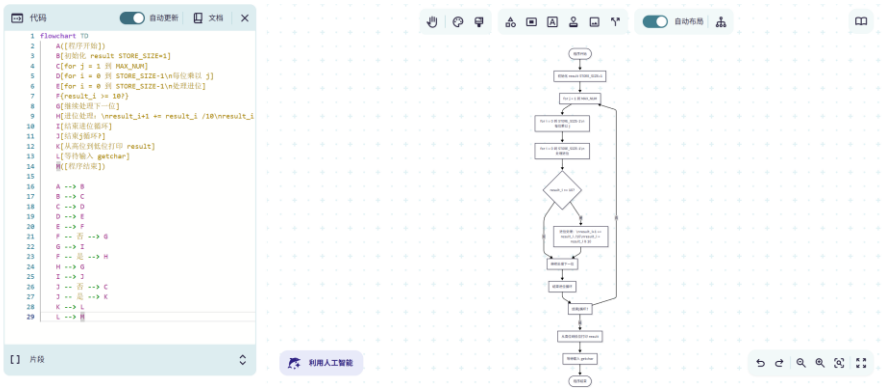
1.开始:
程序开始执行。
2.初始化 result[]:
result[] 数组被初始化，其中 result[0] 设置为 1，表示最小乘积单位。其余的数组元素被初始化为 0。
3.外层循环（对于从 1 到 MAX_NUM 的每个整数 j）:
这是一个从 1 到 1000 的循环，用于逐步计算阶乘。
4.内层循环（对于 result[] 中的每个元素）:
这个循环遍历 result[] 数组中的每一位，与当前的 j 相乘。
5.乘法操作:
在内层循环中，数组的每一个元素 result[i] 与 j 相乘。



6.检查是否需要处理进位:
在乘法操作后，检查每位是否大于或等于 10，以确定是否需要进位。
7.处理进位:
如果 result[i] 大于等于 10，则将进位部分（result[i] / 10）加到下一位 result[i+1] 上，然后更新 result[i] 为 result[i] % 10。
8.打印结果:
从 result[] 数组的最后一个非零元素开始向前打印，因为数组是以逆序方式存储的数字。
9.等待输入:
调用 getchar() 函数暂停程序，等待用户输入，这样用户可以看到屏幕上打印的结果。
10.结束:
程序执行完毕，结束运行。

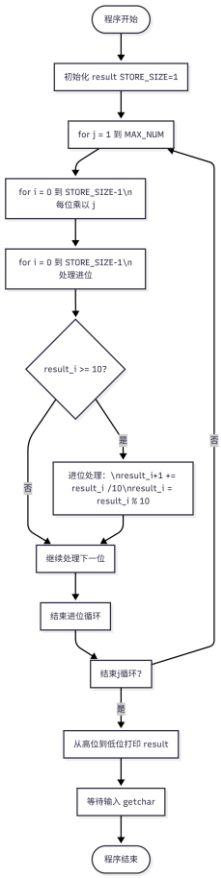


使用 AI 平台分析代码并生成 Mermaid 代码绘图：



```
flowchart TD
    A([程序开始])
    B[初始化 result STORE_SIZE=1]
    C[for j = 1 到 MAX_NUM]
    D[for i = 0 到 STORE_SIZE-1\n每位乘以 j]
    E[for i = 0 到 STORE_SIZE-1\n处理进位]
    F{result_i >= 10?}
    G[继续处理下一位]
    H[进位处理：\nresult_i+1 += result_i /10\nresult_i = result_i % 10]
```

```
    I[结束进位循环]
    J[结束 j 循环?]
    K[从高位到低位打印 result]
    L[等待输入 getchar]
    M([程序结束])

    A --> B
    B --> C
    C --> D
    D --> E
    E --> F
    F -- 否 --> G
    G --> I
    F -- 是 --> H
    H --> G
    I --> J
    J -- 否 --> C
    J -- 是 --> K
    K --> L
    L --> M
```



高清图可查看附件 Mermaid.jpg

实验结论： 我的流程图采用传统程序流程图规范，通过椭圆（开始 / 结束）、矩形（处理）、菱形（判断），严格遵循代码的线性执行顺序，更接近代码执行的顺序。AI 生成的流程图会更偏向于模块化、概括化，把循环和处理步骤合并成模块，判断框也更简洁，突出逻辑块而不是逐行的细节。并且 AI 生成的流程图代码会出错，绘制的流程图也并非使用传统的规范画图方法，如部分判断处没使用菱形。 实际开发应用中，可以借助 AI 工具，有效提高效率，但是也要批判性地对 AI 生成的内容进行选择使用。
指导教师批阅意见：
成绩评定： 指导教师签字： 年　　月　　日
备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。