

【注意:】

- 1、本次作业不允许使用尚未讲授过的任何后续课程中的知识点，包括但不限于结构体、类等相关概念!!!
- 2、除明确要求外，已学过的知识中，**不允许**使用 goto、**不允许**使用全局变量（不包括宏定义和只读变量），**不允许**使用 C++ 的 string 变量
- 3、cstdio、cmath、cstring 中的系统函数**可以**直接使用，包括课上未介绍过的，具体可自行查阅相关资料
- 4、除明确要求外，所有 cpp 源程序不允许使用 scanf/printf 进行输入/输出
- 5、多编译器下均要做到“0 errors, 0 warnings”
- 6、部分题目要求 C 和 C++ 两种方式实现，具体见网页要求
- 7、输出为浮点数且未指定格式的，均要求为 double 型，C++ 为 cout 缺省输出，C 为 %lf 的缺省输出
- 8、认真阅读格式要求及扣分说明!!!

【本次作业特别要求:】

- 1、本章所有作业都必须用指针（即使出现数组，也必须是指针法访问）
- 2、一维数组，除定义变量外，函数的形参及程序的执行语句不允许出现 [] 形式
- 3、指针在一维数组的元素间移动时，要体现出指针变量自身的变化（具体要求参考 060001 课件，要求为类似 P. 59 的左侧形式，如果写成右上或右下则得分为 0!!!）

【输出格式要求:】

- 1、为方便机器自动判断正确性，作业有一定的输入输出格式要求（但不同于竞赛的无任何提示）
- 2、每个题目见具体说明，必须按要求输入和输出，不允许有偏差
- 3、没有特别说明的情况下，最后一行有效输出的最后有一个 endl

补充:

- 4、用指针访问一维字符数组的方式实现下列函数

函数原型	功能说明	返回值
int tj_strlen(const char *str);	求字符串 str 的长度	字符串长度
char *tj_strcat(char *s1, const char *s2);	将字符串 s2 追加到 s1 后面，含 \0	新的 s1
Char *tj_strncat(char *s1, const char *s2, const int len);	将字符串 s2 的前 len 个字符追加到 s1 后面，并添加 \0 ★ 若 len 比 s2 的长度大，则追加整个 s2 即可 (含 \0)	新的 s1

<code>char *tj_strcpy(char *s1, const char *s2);</code>	将字符串 s2 复制到 s1 中, 覆盖 s1 中原内容, 复制时包含\0	新的 s1
<code>char *tj_strncpy(char *s1, const char *s2, const int len);</code>	将字符串 s2 的前 len 个字符复制到 s1 中, 复制时不含\0 ★ 若 len 比 s2 的长度大, 复制 s2 长度个字符即可(不含\0)	新的 s1
<code>int tj_strcmp(const char *s1, const char *s2);</code>	比较字符串 s1 和 s2 的大小, 英文字母要区分大小写	相等为 0, 不等则为第 1 个不相等字符的 ASCII 差值
<code>int tj_strcasecmp(const char *s1, const char *s2);</code>	比较字符串 s1 和 s2 的大小, 英文字母不分大小写 ★ 例: <code>tj_strcasecmp("abc", "ABZ");</code> <code>tj_strcasecmp("ABC", "abz");</code> 均返回-23 ★ 例: <code>tj_strcasecmp("abZ", "AB[");</code> <code>tj_strcasecmp("abz", "AB[");</code> 均返回 31	相等为 0, 不等则为第 1 个不相等字符的 ASCII 差值 ★ 若不相等处字符是不同的大小写字母, 则统一转换为小写后比较 ★ 若不相等处是大写字母和其它字符, 则返回对应小写字母和其它字符的差值
<code>int tj_strncmp(const char *s1, const char *s2, const int len);</code>	比较字符串 s1 和 s2 的前 len 个字符的大小, 英文字母要区分大小写 ★ 若 len 大于 s1/s2 中长度短的串, 则比较到短串的\0 即结束	相等为 0, 不等则为第 1 个不相等字符的 ASCII 差值
<code>int tj_strcasencmp(const char *s1, const char *s2, const int len);</code>	比较字符串 s1 和 s2 的前 len 个字符的大小, 英文字母不分大小写 ★ 长度要求同 <code>tj_strncmp</code> ★ 大小写要求同 <code>tj_strcasecmp</code>	相等为 0, 不等则为第 1 个不相等字符的 ASCII 差值
<code>char *tj_strupr(char *str);</code>	将字符串 str 中所有小写字母均转为大写, 其它字符不变, 转换后放在原串中	转换后的串
<code>char *tj_strlwr(char *str);</code>	将字符串 str 中所有大写字母均转为小写, 其它字符不变, 转换后放在原串中	转换后的串
<code>int tj_strchr(const char *str, const char ch);</code>	在字符串 str 中寻找字符 ch 第 1 次出现的位置, 顺序是从左到右	找到: 返回 1-n(位置从 1 开始), 未找到则返回 0
<code>int tj_strstr(const char *str, const char *substr);</code>	在字符串 str 中寻找字符串 substr 第 1 次出现的位置, 顺序是从左到右	找到: 返回 1-n(位置从 1 开始), 未找到则返回 0

int tj_strchr(const char *str, const char ch);	在字符串 str 中寻找字符 ch 第 1 次出现的位置，顺序是从右到左 ★ 例：tj_strchr("abcdab", 'a') 返回 5	找到：返回 1-n(位置从 1 开始)，未找到则返回 0
int tj_strrstr(const char *str, const char *substr);	在字符串 str 中寻找字符串 substr 第 1 次出现的位置，顺序是从右到左 ★ 例：tj_strrstr("abcdab", "ab") 返回 5	找到：返回 1-n(位置从 1 开始)，未找到则返回 0
char *tj_strrev(char *str);	字符串反转，放入原串中	反转后的串

- 本题要求：**
- 1、**不允许**使用任何系统函数（strlen、strcpy 等），**不允许**使用 C++ 的 string 类
 - 2、可以用自己定义的函数（例如在其它 tj_**函数中调用 tj_strlen）
 - 3、函数实现时不必考虑空间不够的情况（空间由调用函数保证）
 - 4、给出 6-b4.h、6-b4-main.cpp、6-b4-sub.cpp 三个文件共同形成一个可执行文件，6-b4.h 用于函数声明，6-b4-main.cpp 是测试用例，这两个文件**不准改动，无需提交**；列表中所有函数的具体实现均在 6-b4-sub.cpp 中，每个函数实现时有具体要求，必须按要求实现，提交时只需提交此文件即可
 - 5、要求使用指针的自增/相减/比较等运算，**不能**简单地将 str[i] 改为 *(str+i) 的形式，否则**本题得分为 0**
（具体要求参考 060001 课件，要求为 P.60 的左侧形式，如果写成右上或右下则得分为 0!!!）
 - 6、整个程序中**不准**用任何形式的全局变量及静态局部变量；在上表列出的所有函数中，均**不准**定义任何数组形式的自动变量
 - 7、再次提醒，指针为 NULL 和空字符串是两个完全不同的概念，因此对 NULL 有特殊要求，详见 main 函数
 - 8、针对 tj_strcmp/tj_strcasecmp/tj_strncmp/tj_strcasncmp 这四个比较函数，如果 s1/s2 中一个为 NULL 一个为非 NULL，则返回 1 或 -1 即可

5、写一个用矩形法求定积分的通用函数，分别求 $\int_0^1 \sin x dx$ 、 $\int_{-1}^1 \cos x dx$ 、 $\int_0^2 e^x dx$ 的值

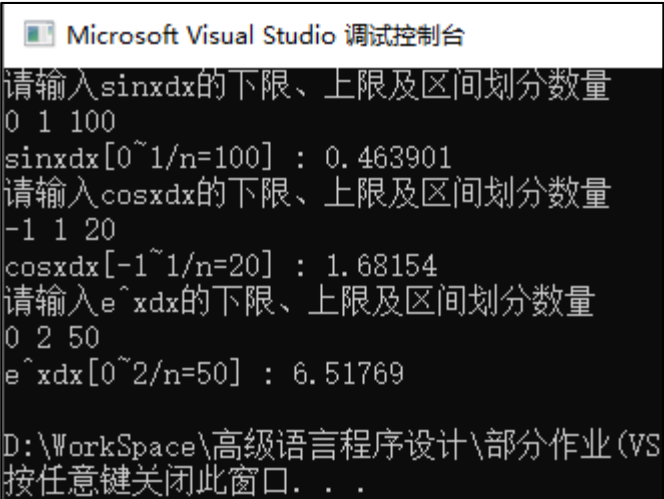
【要求:】给出下面的程序框架，用 definite_integration 函数求三种不同的定积分

```
返回类型 definite_integration (形参表)
{
    通用函数的实现部分 (不允许任何选择形式[if/switch/条件表达式]语句出现)
}
int main()
{
    int n;
    double low, high, value;
    输入上下限及划分数目 (不考虑输入错误)
    value = definite_integration(sin, 其它参数); //sin 为 cmath 中已有函数
    cout << sinxdx 的值 << endl;

    输入上下限及划分数目 (不考虑输入错误)
    value = definite_integration(cos, 其它参数); //cos 为 cmath 中已有函数
    cout << cosxdx 的值 << endl;

    输入上下限及划分数目 (不考虑输入错误)
    value = definite_integration(exp, 其它参数); //exp 为 cmath 中已有函数
    cout << e^xdx 的值 << endl;

    return 0;
}
```



```
Microsoft Visual Studio 调试控制台
请输入sinxdx的下限、上限及区间划分数目
0 1 100
sinxdx[0~1/n=100] : 0.463901
请输入cosxdx的下限、上限及区间划分数目
-1 1 20
cosxdx[-1~1/n=20] : 1.68154
请输入e^xdx的下限、上限及区间划分数目
0 2 50
e^xdx[0~2/n=50] : 6.51769

D:\Workspace\高级语言程序设计\部分作业(VS
按任意键关闭此窗口...
```

注：1、输出为英文冒号，前后各一个空格

2、本题为 C 方式

3、提供 6-b5-demo.exe 供参考（demo 考虑了输入错误，不需要实现）

4、矩形计算取右值，输出为正常浮点数（double）格式（和 demo 保持小数点后 2 位相同即可，本题的关键不是值的准确性）

6、已知月份存储在指针数组中，形式如 `const char *month[]={..., "March", ...}`，要求键盘输入代表月份的整数，输出对应的英文月份名

本题要求：1、英文月份名为全称，首字母大写

2、任何输入错误，包括非整数、数据范围错误等均输出 “Invalid”

3、month 数组必须为指针数组，数组大小、是否 const、月份的排列顺序等按需决定

4、除了下面给出的框架外，按一句一行（即：1 个 if-else 至少 4 行）的规则，填入语句行数不超过 8 行

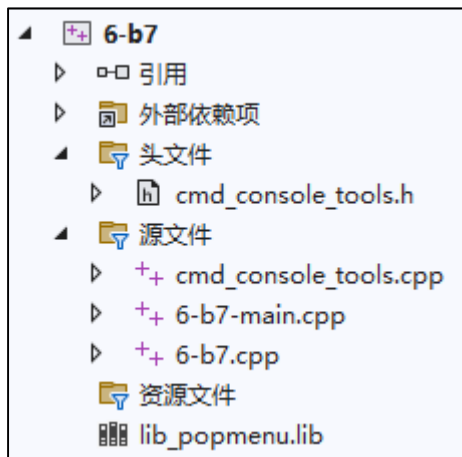
a) 不允许逗号表达式

b) **必须**用 “cin>>整形变量” 的方式读入整数形式的月份，而且必须要判断 cin 的状态（即 false 给出 Invalid）

```
#include <iostream>
using namespace std;
int main()
{
    //除已有6行外，填入语句行数不超过8行（不含注释）
    return 0;
}
```

5、提供 6-b6-demo.exe 供参考

7、一个菜单的各项存储在指针数组中，要求用弹出式菜单的方式显示菜单，箭头键上下移动、回车键选择、ESC 键放弃，并打印选择的项数
对应项目的建立方法：



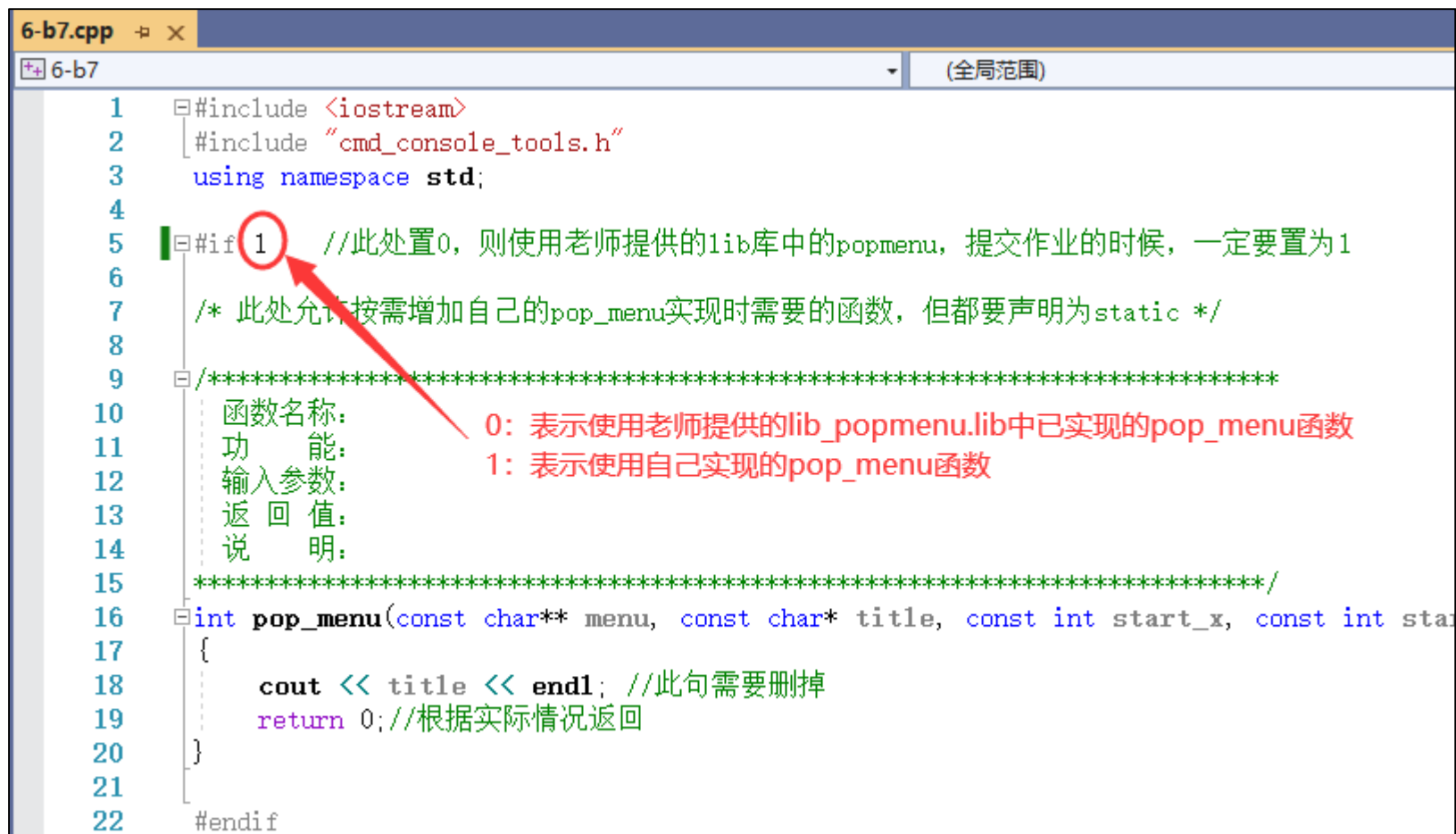
建立如图所示的 6-b7 项目

- 将附件给出的三个文件去掉前缀后加入项目中（添加现有项）
- cmd_console_tools.h 和 cmd_console_tools.cpp 用汉诺塔作业给出的即可
- 项目中一共是 5 个文件

关于静态链接库的简单说明：

- 1、*.lib 是由源代码编译而成的静态链接库文件，用户可以调用其中的函数，但是看不到函数的具体实现过程，适用于需要给他人提供工具函数的使用而不提供源代码的应用场景
- 2、附件给出的 lib_popmenu.lib 包含了老师提供的 pop_menu 函数的实现，可直接调用，方法见下

不同 pop_menu 函数的调用方法：



```
6-b7.cpp 6-b7 (全局范围)
1  #include <iostream>
2  #include "cmd_console_tools.h"
3  using namespace std;
4
5  #if 1 //此处置0，则使用老师提供的lib库中的popmenu，提交作业的时候，一定要置为1
6
7  /* 此处允许按需增加自己的pop_menu实现时需要的函数，但都要声明为static */
8
9  /*****
10     函数名称：
11     功    能：
12     输入参数：
13     返 回 值：
14     说    明：
15     *****/
16  int pop_menu(const char** menu, const char* title, const int start_x, const int sta
17  {
18     cout << title << endl; //此句需要删掉
19     return 0; //根据实际情况返回
20  }
21
22  #endif
```

0: 表示使用老师提供的lib_popmenu.lib中已实现的pop_menu函数
1: 表示使用自己实现的pop_menu函数

本题要求：

- 1、pop_menu 函数的参数解释
 - const char** menu : 指针数组形式的菜单项，最后一个必须是 NULL
 - const char* title : 弹出式菜单的标题栏，显示在最上方，居中
 - const int start_x : 左上角 x 坐标（理解同之前 cmd_console_tools）
 - const int start_y : 左上角 y 坐标（理解同之前 cmd_console_tools）
 - const int high : 弹出式菜单同时显示的项数（不含上下边框，即总高度要+2）
 - const int width : 弹出式菜单每项的字符数（不含左右边框，即总宽度要+4）
 - const int bk_color : 背景色（理解同之前 cmd_console_tools）
 - const int fg_color : 前景色（理解同之前 cmd_console_tools）
- 2、半个汉字乱码问题：
 - 菜单边框为汉字，占两个字节，因此菜单的宽度一定是偶数（具体可参考选做作业的附录）
 - 因为边框的影响，如果标题栏的字符数为奇数，后面多补一个空格
例：width=10,title="abc"，则显示 title 为"abc "，否则会有半个汉字的乱码
 - 因为边框的影响，如果标题栏相对于 start_x 的起始位置算出来是奇数，则减一
例：width=10,title="abcd"，则应该从 start_x+5 开始显示"abcd"，实际应从 start_x+4 开始，否则会有半个汉字的乱码
 - 因为边框的影响，如果某菜单项最后一个字符是某个汉字的前半字节，则输出空格，否则会有半个汉字的乱码出现
 - 以上所说的半个汉字乱码，在实际完成过程中自行体会
- 3、main 函数的第三个测试用例，折返到下一行的右边框不存在，这是坐标越界导致的问题，不需要解决
- 4、需要按之前的要求设置为旧版控制台
- 5、仅需要 VS2022 实现即可

提交要求：

- 1、只需要提交 6-b7.cpp 一个文件，其余文件不需要提交
- 2、可以任意修改 6-b7-main.cpp 文件，实际测试时会替换本文件（即不要认为测过 main 函数的四种情况就算完全通过了）

【编译器要求：】

		编译器VS	编译器Dev
6-b4-sub.cpp	一组字符串处理函数	Y	Y
6-b5.c	矩形法求定积分	Y	Y
6-b6.cpp	指针数组输出月份	Y	Y
6-b7.cpp	弹出式菜单工具的实现	Y	/

【作业要求:】

- 1、**12月13日前**网上提交本次作业
- 2、每题所占平时成绩的具体分值见网页
- 3、超过截止时间提交作业会自动扣除相应的分数，具体见网页上的说明