2023 年计算机程序设计（智）机试题解析

AI 学辅

1 题目一

1.1 题目信息

学院为鼓励大家多进行体育运动，通过挣积分换礼品方式调动大家体育运动积极性，学院规定每天打卡运动时间为 x：如果运动时间 x 为 0，奖励积分为 $y = 0$[注 1]；如果运动时间 x 大于 0 小于等于 60 分钟，奖励积分为 $y = 2x + 10$；如果运动时间 x 大于 60 分钟小于等于 120 分钟，奖励积分为 $y = x^2 + 2x + 10$；如果运动时间 x 大于 120 分钟小于等于 600 分钟，奖励积分为 $y = 5x - 30$；如果运动时间 x 大于 600 分钟小于等于 1440 分钟（24 小时），奖励积分为 $y = x/5 - 200$。其它不合理的输入情况均会输出 `input error`。学院要求各班体育委员统计大家每天积分，请帮体育委员写个小程序，输入学生的运动时间，该小程序立刻算出学生本次运动的奖励积分。

样例

输入	`25`	**输出**	`60`
输入	`sdfsd`	**输出**	`input error`

1.2 解题思路

输入处理 考虑到不能根据整型变量的值来判断输入是否为整数，我们可以将输入流的数据存储到 `string` 类变量中，再通过检查 `string` 中的字符是否都为数字来判断输入是否为正整数。若输入为正整数，通过 `stoi` 函数将 `string` 类变量转换为整型（在 `std` 命名空间中）。至于更高级的方法，可以直接通过检查输入流的返回值来判断输入是否合法（`if(!(cin>>x))`）。

积分计算 使用 `if-else` 语句逐步判断即可，x 不在 $[0, 1440]$ 内时输出 `input error`。注意输出 `input error` 后要 `return 0`，终止后续代码的执行。

输出结果 直接使用 `cout` 输出即可。

1.3 代码实现

```
#include <iostream>
```

1. ↑ 因**实际表现无对应描述**，我们对其进行了补充。

```cpp
#include <string>
using namespace std;

int main()
{
    string str;
    int x, y;
    cin >> str;
    for (char c : str)
        if (!isdigit(c)) {
            cout << "input error";
            return 0;
        }
    x = stoi(str);

    if (x<0||x>1440) {  // 防止越界
        cout << "input error";
        return 0;
    }
    if (x==0) y = 0;
    else if (x<=60) y = 2*x+10;
    else if (x<=120) y = x*x+2*x+10;
    else if (x<=600) y = 5*x-30;
    else if (x<=1440) y = x/5-200;
    cout << y;
}
```

2 题目二

2.1 题目信息

某学生编写了一个简单的即时通讯软件，为了防止消息在传输中被窃取，将对每次聊天内容进行简单加密（每次消息发送最多支持 199[注 2]字符，否则被分成下一条发送）：统计用户输入字符数并输出；对用户输入的内容中包括英文字母的首先进行大小写转换并加 2；对用户输入的内容中包含数字的部分进行加 5 操作，然后对结果模 10[注 3]；其它不做操作。请编程实现该功能并输出加密内容（下附 ASCII 码表）。注意：

2. ↑ 此处原文为 **200**，因**占位符\0 的存在**，我们对其进行了修正。

3. ↑ 此处原文为**如果超过 10，对结果减 10**，因**描述与实际不符**，我们对其进行了修正。

该软件的主函数已经写好，仅补充 MyInputFun，MyCodeFun 及 MyShowFun 三个子函数，实现软件功能。

```
void main() {
    char* str1 = NULL;
    char* str2 = NULL;
    char strMesg[200];
    char strMesg_code[200];

    str1 = MyInputFun(strMesg);
    str2 = MyCodeFun(strMesg_code, str1);
    MyShowFun(str2);
}
```

ASCII 打印字符

0010		0011		0100		0101		0110		0111		
2		3		4		5		6		7		
十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	ctrl
32		48	0	64	@	80	P	96	`	112	p	
33	!	49	1	65	A	81	Q	97	a	113	q	
34	"	50	2	66	B	82	R	98	b	114	r	
35	#	51	3	67	C	83	S	99	c	115	s	
36	$	52	4	68	D	84	T	100	d	116	t	
37	%	53	5	69	E	85	U	101	e	117	u	
38	&	54	6	70	F	86	V	102	f	118	v	
39	'	55	7	71	G	87	W	103	g	119	w	
40	(	56	8	72	H	88	X	104	h	120	x	
41	)	57	9	73	I	89	Y	105	i	121	y	
42	*	58	:	74	J	90	Z	106	j	122	z	
43	+	59	;	75	K	91	[	107	k	123	{	
44	,	60	<	76	L	92	\	108	l	124	|	
45	-	61	=	77	M	93	]	109	m	125	}	
46	.	62	>	78	N	94	^	110	n	126	~	
47	/	63	?	79	O	95	_	111	o	127	Δ	Back space

样例

输入 I am XJTUer; class 2301 2302;

输出 k CO zlvwGT; ENCUU 7856 7857;

29

2.2 解题思路

MyInputFun 题目要求读取一行带空格的字符串，可以使用 `cin.getline` 函数为 `char[]` 类型赋值，随后返回其指针。

MyCodeFun 字符的加减就是其 ASCII 码的加减，因此只需要遍历字符串（终止于`'\0'`），对每个字符进行判断并加密即可。对于字母的小写转大写，计算其与`'a'` 的差值，再加上`'A'` 就是其对应的大写字母（大写转小写同理），最后加上偏移量 2 得加密字符；对于数字，计算其与`'0'` 的差值，加上 5 的偏移量再模 10 即得加密字符；其他字符不变。

MyShowFun 依次输出加密后的字符串，再输出总字符数即可。

2.3 代码实现

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

char* MyInputFun(char s[]) {
    cin.getline(s, 200);  // 读取一整行（上限200个字符）
    return s;
}

char* MyCodeFun(char s2[], char* s1) {
    for(int i=0; s1[i]!='\0'; i++) {
        if('a'<=s1[i]&&s1[i]<='z')  // 小写转大写加2
            s1[i] = 'A'+s1[i]-'a'+2;
        else if('A'<=s1[i]&&s1[i]<='Z')  // 大写转小写加2
            s1[i] = 'a'+s1[i]-'A'+2;
        else if('0'<=s1[i]&&s1[i]<='9')  // 数字加5取模
            s1[i] = '0'+(s1[i]-'0'+5)%10;
        else s1[i] = s1[i];  // 其他字符不变
    }
    return s1;
}  // 这里没有用到s2。如果将加密后的字符存入s2，一定记得在末位后加上'\0'
    （未初始化的字符数组'\0'的位置不确定）

void MyShowFun(char* s) {
    int i=0;
```

```
    for( ; s[i]!='\0'; i++) cout << s[i];
    cout << endl << i;  // 输出字符个数
}
```

3 题目三

3.1 题目信息

请帮某学生实现一个成绩分析小程序，首先由用户输入 4 名学生的学号、高等数学、大学英语及程序设计课程的成绩，并将学号及 3 门课的成绩存入结构体数组，结构体成员包括学生学号及该学生的 3 门课程的成绩。这些信息从键盘输入（成绩范围为 $[0, 100]$，学生学号为 23001、23002、23003…等形式）。接着计算所有学生平均分及 GPA（高等数学、大学英语及程序设计的学分分别为 5、4、3，GPA 的算法为所有课程的成绩乘学分累加除以所有学分之和；再将学生的学号、平均分及 GPA 存放于另一个结构体数组并打印输出；最后统计这些学生的高等数学、大学英语及程序设计课程的平均分并打印输出。

（注意务必按照题目中的结构体进行编程，无结构体实现则扣分）

样例

输入 23001 56 85 62 23002 45.2 56.1 25.6 23003 95.5 96.3 94.2
23004 56.5 95.6 94.5

输出 23001 67.6667 67.1667
23002 42.3 43.9333
23003 5.3333 95.4417
23004 82.2 79.0333
63.3 83.25 69.075

（输出数字之间的空位为 2 个空格）

3.2 解题思路

这道题并不难，主要考察对结构体的基本运用。先按题目要求定义两个结构体数组，一个包含学号和 3 门课的成绩，另一个包含学号、平均分和 GPA（这里使用 `string` 类型存储学号，就不用担心字符串大小的问题）；使用循环读取每个学生的学号和 3 门课的成绩并存入第一个结构体数组中；使用循环遍历第一个结构体数组，计算每个学生

的平均分和 GPA 并存入第二个结构体数组中（GPA 就是成绩对学分的加权平均数）；使用循环遍历第二个结构体数组，输出每个学生的学号、平均分和 GPA（注意输出小数时可能要求保留位数，这里不需要）；最后统计所有学生 3 门课的平均分并输出，这一步可以在先前的循环中计算。

3.3 代码实现

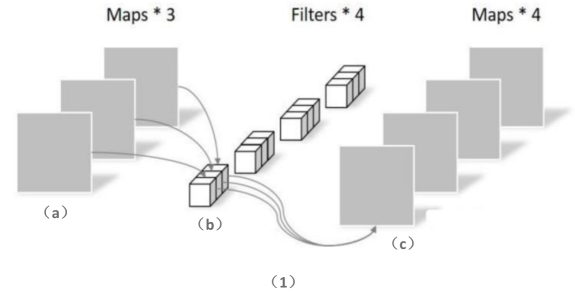
```
#include <iostream>
#include <string>
using namespace std;

struct STU1
{
    string uid;
    float s1, s2, s3;
} stu1[4];  // 分号不能少

struct STU2
{
    string uid;
    float avg, gpa;
} stu2[4];

int main() {
    int i;
    float sum1=0, sum2=0, sum3=0;  // 每门课的总分
    for(i=0;i<4;i++) cin>>stu1[i].uid>>stu1[i].s1>>stu1[i].s2>>stu1[i].s3;
    for(i=0;i<4;i++) {
        stu2[i].uid = stu1[i].uid;
        stu2[i].avg = (stu1[i].s1 + stu1[i].s2 + stu1[i].s3) / 3;
        stu2[i].gpa = (5*stu1[i].s1 + 4*stu1[i].s2 + 3*stu1[i].s3) / 12;
    }
    for(i=0;i<4;i++) {
        sum1+=stu1[i].s1; sum2+=stu1[i].s2; sum3+=stu1[i].s3;
        cout<<stu2[i].uid<<"  "<<stu2[i].avg<<"  "<<stu2[i].gpa<<endl;
    }
    cout<< sum1/4 << "  " << sum2/4 << "  " << sum3/4;
}
```

4 题目四

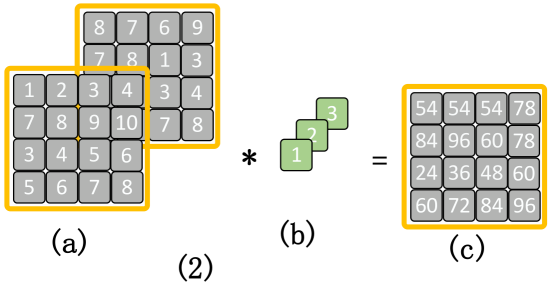
4.1 题目信息

当前推动人工智能发展进入新的一轮高潮的一个重要因素之一是深度学习的突破性进展，而深度学习中卷积神经网络起到的作用尤为重要，卷积神经网络中，深度可分离卷积是一种特殊而常用的轻量化卷积结构，请基于下列提示，自动动手写出一个类似的深度可分离卷积的运算，按照下面说明编程矩阵运算：

如下图所示，Maps 包含若干个 $N \times N$ 矩阵 (a)，Filters 是 $1 \times N$ 矩阵 (b)，$N \times N$ 矩阵的个数代表通道数，运算时若干个 $N \times N$ 矩阵逐点与 $1 \times N$ 矩阵乘加，最后生成一张 Maps(c)。本题简化计算如图 (2)：输入为两个 4×4 矩阵 (a)，两个 4×4 矩阵分别与 1×4 矩阵 (b) 相乘相加，即 (a) 中最前面矩阵的第一个元素 1 与矩阵 (b) 中元素分别相乘加：即 $1*1+1*2+1*3=6$；(a) 中最后面矩阵的第一个元素 8 与矩阵 (b) 中元素分别相乘加：即 $8*1+8*2+8*3=48$；前面矩阵运算结果 6 再与后面矩阵运算结果 48 相加，得最终结果矩阵 (c) 的第一个元素的值 54。



(1)



(2)

样例

输入
```
1 2 3
1 2 3 4 7 8 9 10 3 4 5 6 5 6 7 8
8 7 6 9 7 8 1 3 1 2 3 4 5 6 7 8
```
输出
```
54  54  54  78
84  96  60  78
24  36  48  60
60  72  84  96
```
（输出数字之间的空位为 2 个空格）

4.2 解题思路

输入处理 由于矩阵尺寸既定，直接定义两个 4×4 的二维数组和一个 3 维向量即可。注意到矩阵 (c) 的元素就是两个 4×4 矩阵的对应位置元素之积再乘上矩阵 (b) 的元素总和，因此可以将两个 4×4 矩阵展开成 16 维向量并在读取矩阵 (b) 时直接计算其元素总和以简化运算（这题的运算只是个特例，不是真正的卷积运算）。

矩阵运算 直接逐位遍历计算两个 16 维向量的对应位置元素之积，再乘以矩阵 (b) 的元素之和即可。

输出结果 题目仅要求输出矩阵 (c) 的元素，因此可以直接在循环中使用 `cout` 输出。由于每输出 4 个元素后要换行，可以引入一个计数器，每输出一个元素后计数器加 1，当计数器为 4 的倍数时输出换行符，否则输出两个空格。

4.3 代码实现

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

int main() {
    float mat1[16], mat2[16], b, b_sum=0;
    int x, i, count=0;
    for(i=0; i<3; i++) {cin >> b; b_sum += b;}  // 计算b的元素和
    for(i=0; i<16; i++) cin >> mat1[i];
    for(i=0; i<16; i++) cin >> mat2[i];
```

```
    for(i=0; i<16; i++) {
        cout << (mat1[i] + mat2[i]) * b_sum;
        if(++count%4 == 0) cout << "\n"; else cout << "   ";
        // 或者使用三元运算符 cout << ((++count%4==0) ? "\n" : "   ");
    }
}
```