**作业I** 等级：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

学号：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 姓名:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[F1]循环指令loop在保护方式（32位代码段）下采用寄存器\_\_\_\_\_\_\_作为循环计数器，在实方式下，采用寄存器\_\_\_\_\_\_\_\_作为循环计数器。

[F2]循环指令loop的转移范围是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_，以其自身起点为基准。

[F3]循环指令loop的功效相当于指令\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

[F4]配合循环指令的条件转移指令是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

[S1]以下关于循环实现方法的叙述，正确的是（多选）？

（A）利用循环指令可以方便地编写循环程序片段。

（B）不利用循环指令，也可以编写循环程序片段。

（C）利用条件转移指令，无法实现循环。

（D）利用传送指令，无法实现循环。

[S2]以下关于循环结构的叙述，正确的是（多选）？

（A）循环结构必定有初始化部分。

（B）循环结构必定有循环体部分。

（C）循环结构必定有调整部分。

（D）循环结构必定有分支转移部分。

（E）循环结构必定是先执行循环体，再判断循环条件。

（F）循环结构必定是先判断循环条件，再执行循环体。

[S3]以下关于循环的叙述，错误的是（多选）？

（A）循环是基本的程序结构。

（B）循环嵌套得越多，代码性能越好。

（C）汇编语言中，可以从三重循环的最内层直接跳转到最外层。

（D）汇编语言中，可以从二重循环的外层直接跳转到内层。

[Q1]与条件转移指令相比，采用专门循环指令LOOPE或LOOPNE实现循环有何特点？

[G1]如下子程序的功能是将字符串中的阿拉伯数字符替换为空格字符。请在每一划线处填写一条指令，使得程序完整。

SUBR1:

PUSH EBP

PUSH EBX

MOV EBX, [EBP+8]

NEXT:

MOV AL, [EBX]

JZ OVER

CMP AL,'0'

JB CONTI

CMP AL,'9'

JA CONTI

MOV BYTE PTR [EBX],0x20

CONTI:

OVER:

POP EBX

RET

[R1]设有如下C函数，其功能是计算等差数列之和：

unsigned sum(unsigned a1, unsigned n, unsigned d)

{ unsigned i, sum=0;

for (i=1; i<=n; i++)

sum += a1+(i-1)\*d;

return sum;

}

（1）在VC2010环境中，**禁止编译优化情形下**，可得到如下的目标代码。请说明每条汇编格式指令的作用（可以利用高级语言语句或表达式来描述）。

push ebp

mov ebp, esp

sub esp, 8

mov DWORD PTR [ebp-8], 0

mov DWORD PTR [ebp-4], 1

jmp SHORT LN3@sum

LN2@sum:

mov eax, DWORD PTR [ebp-4]

add eax, 1

mov DWORD PTR [ebp-4], eax

LN3@sum:

mov ecx, DWORD PTR [ebp-4]

cmp ecx, DWORD PTR [ebp+12]

ja SHORT LN1@sum

mov edx, DWORD PTR [ebp-4]

sub edx, 1

imul edx, DWORD PTR [ebp+16]

add edx, DWORD PTR [ebp+8]

add edx, DWORD PTR [ebp-8]

mov DWORD PTR [ebp-8], edx

jmp SHORT LN2@sum

LN1@sum:

mov eax, DWORD PTR [ebp-8]

mov esp, ebp

pop ebp

ret

（2）在VC2010环境中，在编译优化（大小最小化）情形下，可得到如下的目标代码。请说明每条汇编格式指令的作用（可以利用高级语言语句或表达式来描述）。

push ebp

mov ebp, esp

mov edx, DWORD PTR [ebp+12]

xor eax, eax

cmp edx, 1

jb SHORT LN1@sum

xor ecx, ecx

push esi

LL3@sum:

mov esi, DWORD PTR [ebp+8]

add esi, ecx

add ecx, DWORD PTR [ebp+16]

add eax, esi

dec edx

jne SHORT LL3@sum

pop esi

LN1@sum:

pop ebp

ret

（3）在VC2010环境中，在编译优化（速度最大化）情形下，可得到如下的目标代码。请说明每条汇编格式指令的作用（可以利用高级语言语句或表达式来描述）。

push ebp

mov ebp, esp

mov edx, DWORD PTR \_n$[ebp]

xor eax, eax

cmp edx, 1

jb SHORT LN1@sum

push ebx

push esi

mov esi, DWORD PTR \_d$[ebp]

push edi

mov edi, DWORD PTR \_a1$[ebp]

xor ecx, ecx

LL3@sum:

lea ebx, DWORD PTR [ecx+edi]

add eax, ebx

add ecx, esi

dec edx

jne SHORT LL3@sum

pop edi

pop esi

pop ebx

LN1@sum:

pop ebp

ret