


```

double funct(double a, float x, double b, int i)
a in %xmm0, x in %xmm1, b in %xmm2, i in %edi
1  funct:
    The following two instructions convert x to double
2  vunpcklps    %xmm1, %xmm1, %xmm1
3  vcvtps2pd    %xmm1, %xmm1
4  vmulsd      %xmm0, %xmm1, %xmm0    Multiply a by x
5  vcvtsi2sd    %edi, %xmm1, %xmm1    Convert i to double
6  vdivsd      %xmm1, %xmm2, %xmm2    Compute b/i
7  vsubsd      %xmm2, %xmm0, %xmm0    Subtract from a*x
8  ret                                Return

```

三个浮点参数 a 、 x 和 b 通过 XMM 寄存器 $\%xmm0 \sim \%xmm2$ 传递，而整数参数通过寄存器 $\%edi$ 传递。标准的双指令序列用以将参数 x 转换为双精度类型（第 2~3 行）。另一条转换指令用来将参数 i 转换为双精度类型（第 5 行）。该函数的值通过寄存器 $\%xmm0$ 返回。

 **练习题 3.53** 对于下面的 C 函数，4 个参数的类型由 typedef 定义：

```

double funct1(arg1_t p, arg2_t q, arg3_t r, arg4_t s)
{
    return p/(q+r) - s;
}

```


编译时，GCC 产生如下代码：

```

double funct1(arg1_t p, arg2_t q, arg3_t r, arg4_t s)
1  funct1:
2  vcvtsi2ssq    %rsi, %xmm2, %xmm2
3  vaddss        %xmm0, %xmm2, %xmm0
4  vcvtsi2ss     %edi, %xmm2, %xmm2
5  vdivss        %xmm0, %xmm2, %xmm0
6  vunpcklps     %xmm0, %xmm0, %xmm0
7  vcvtps2pd     %xmm0, %xmm0
8  vsubsd        %xmm1, %xmm0, %xmm0
9  ret

```

确定 4 个参数类型可能的组合（答案可能不止一种）。

 **练习题 3.54** 函数 `funct2` 具有如下原型：

```
double funct2(double w, int x, float y, long z);
```

GCC 为该函数产生如下代码：

```

double funct2(double w, int x, float y, long z)
w in %xmm0, x in %edi, y in %xmm1, z in %rsi
1  funct2:
2  vcvtsi2ss     %edi, %xmm2, %xmm2
3  vmulss        %xmm1, %xmm2, %xmm1
4  vunpcklps     %xmm1, %xmm1, %xmm1
5  vcvtps2pd     %xmm1, %xmm2
6  vcvtsi2sdq    %rsi, %xmm1, %xmm1
7  vdivsd        %xmm1, %xmm0, %xmm0
8  vsubsd        %xmm0, %xmm2, %xmm0
9  ret

```

写出 `funct2` 的 C 语言版本。