

ICS'12 Spring Homework 6

- I. In the following code, A and B are constants defined with #define:

```
typedef struct{
    short x[A][B];      /*Unknown constants A and B*/
    int y;
}str1;

typedef struct{
    char array[B];
    int t;
    short s[B];
    int u;
}str2;

void setVal(str1 *p, str2 *q){
    int v1 = q->t;
    int v2 = q->u;
    p->y = v1 + v2;
}
```

Gcc generates the following code for the body of *setVal*:

```
1  movl 12(%ebp), %eax
2  movl 36(%eax), %edx
3  addl 12(%eax), %edx
4  movl 8(%ebp), %eax
5  movl %edx, 92(%eax)
```

What are the values of A and B? (The solution is unique.)

- II. The following declaration defines a class of structures for use in constructing binary trees:

```
typedef struct ELE *tree_ptr;
struct ELF{
    long val;
    tree_ptr left;
    tree_ptr right;
};
```

For a function with the following prototype:

```
long trace(tree_ptr tp);
```

GCC generates the following x86-64 code:

```
1  trace:
   tp in %rdi
2   movl $0, %eax
3   testq %rdi, %rdi
4   je .L3
5   .L5:
6   movq (%rdi), %rax
7   movq 16(%rdi), %rdi
8   testq %rdi, %rdi
9   jne .L5
10  .L3:
11  rep
12  ret
```

A. Generate a C version of the function, using a while loop.

B. Explain in English what this function computes.

III. Writing out the following definitions, you can use the variable **a**.

Examples:

An integer : int a;

*A pointer to an integer : int *a;*

1) A pointer to a pointer to an integer:

2) An array of 10 pointers to integers:

3) A pointer to an array of 10 integers:

4) A pointer to a function that takes an integer argument and returns
an integer:

5) An array of 10 pointers to functions that take an integer argument
and return a pointer to an integer:

IV. The following code transposes the elements of an $M \times M$ array, where

M is a constant defined by `#define`:

```
void transpose(Marray_t A) {
    int i, j;
    for(i=0; i<M; i++)
        for(j=0; j<i; j++) {
            int t = A[i][j];
            A[i][j] = A[j][i];
            A[j][i] = t;
        }
}
```

When compiled with optimization level `-O2`, gcc generates the following code for the inner loop of the function:

```
1  .L3:
2    movl (%ebx), %eax
3    movl (%esi,%ecx,4), %edx
4    movl %eax, (%esi,%ecx,4)
5    addl $1, %ecx
6    movl %edx, (%ebx)
7    addl $52, %ebx
8    cmpl %edi, %ecx
9    jl .L3
```

A. What is the value of M ?

B. What registers hold program values i and j ?

C. Write a C code version of transpose that makes use of the optimizations that occur in this loop. Use the parameter M in your code rather than numeric constants.