Assignment #2: Subprograms and Parameter passing

Structure of Programming Languages
Fall 2008
Due on Thursday, September 25, 2008

Problem 1 a. What are the means of information exchange between subprograms? Explain.

Problem 1b. What do you understand by import and export mechanism of variables? What are the advantages of export and import mechanism? Explain.

Problem 1c. Compare the advantages and disadvantages of Call by Value and Call by Reference.

Problem 1d. Compare the advantages and disadvantages of Call by value result and call by reference. Explain using an example not similar to given in the book or in the classroom.

Problem 1e. What do you understand by side-effect? How is side-effect caused? Explain.

Problem 1f. What problems can be caused by side-effect. Explain using an example not given in the book or in the classroom.

Problem 1g. Explain call by name and the operation that takes place during call by name using a simple example not given in the book or in the classroom.

Problem 1h. What do you understand by aliasing? How does aliasing contribute to side effect? Explain.

Problem 2. show the value of the variables for the following program during both the calls to function messy. Assume that & in the actual parameter passes the reference, # is the actual parameter passes the parameter using call by value result, $ in the actual parameter is used to pass the parameter using call by result.. The absence of any annotation denotes call by value.

```
Program main ()
int i, j, k, a[6];
{
i = 0; j = 0; k = 2;
for (i = 1; i =< 5; i++) a[i] = 10;
messy(a[1], @a[2], @j, @j, #a[3], $k);
messy(a[2], @a[3], @j, @k, #a[1], $a[1]);

 }
```

```
void messy(int A, *B, *C, *D, E, F)
{
   A = *B + *C;
   *B = *D + E + F;
   *C = A + E;
  *D = *C - *D;
  E = *B + *C;
  F = E + A;
}
```

Problem 3. The following program uses Call by Name. Show the execution trace. Show the program main after the substitution of formal parameters by actual parameters, and give the value of all the variables.

```
program main ( )
int i, j, k, a[10];
{
for (i = 0; i =< 9, i++) a[i] = 10 * i;
j = 4; k = 1; i = 2;
messy_name(i, k, j, a[i], a[I +1]);
}
```

```
void messy_name(int m, b, c, d, e);
int k;
{
m = b; k = 2; a[k] = c; a[m] = d; e = 15;
}
```