Assignment

[redacted]

Chapter 9, Problem Set

Ques. 5 Consider the following program in C syntax:

```
void swap (int a, int b)
{
      int temp ;
      temp = a ;
      a = b ;
      b = temp ;
}
void main ()
{
   int value = 2, list[5] = {1, 3, 5, 7, 9};
   swap( value, list [0]);
   swap ( list [0], list [1]);
   swap ( ~~dist~~ value, list [value]);
}
```

For each of the following parameter - passing methods, what are all of the values of the variables value and list after each of three calls to swap?

(a) Passed by value
This is a parameter - passing method, where values of actual parameters are copied to formal parameters of function. These two parameters (actual and formal) have different locations. Therefore, if any changes are made to formal parameters, nothing can reflected in actual parameters.

So, after the execution of the subprogram, values of actual parameters will remain same.

value = 2
list [5] = {1, 3, 5, 7, 9};

(b) Passed by ~~parameter~~ reference
This a parameter - passing method, where formal and actual parameters share same location. If any change is made in formal parameter, the same will be reflected in actual parameter.

1) Before execution of subprogram,
value = 2, list[5] = {1, 3, 5, 7, 9}

2) After implementing first swap function
value = 1, list[5] = {2, 3, 5, 7, 9}

3) After implementing second swap function
value = 1, list[5] = {3, 2, 5, 7, 9}

4) After implementing final swap function.
value = 2, list [5] = {3, 1, 5, 7, 9}

(c) Passed by value-result :
As the function executes, pass by value-result copies the values of actual parameters in to formal parameters. So, changes made in formal parameters are reflected to actual parameters.

1) Before execution of sub-program
value = 2 , list[5] = {1, 3, 5, 7, 9}

2) After implementing first swap function
value = 1 , list[5] = {2, 3, 5, 7, 9}

3) After implementing second swap function
value = 1 , list[5] = {3, 2, 5, 7, 9}

4) After implementing final swap function
value = 2 , list[5] = {3, 1, 5, 7, 9}

Ques.7 Consider the following ~~syntax in~~ program written in C syntax :

```
void fun ( int first, int second)
{
    first += first;
    second += second ;
}

void main ()
{
    int list [2] = {1, 3} ;
    fun ( list [0] , list [1]);
}
```

DATE
PAGE

For each of following parameter-passing methods, what are the values of list array after execution?

(a) Passed by value.
A:: When 'passed by value' method is executed, a separate copy of actual parameter is made inside function. If there is a modification made inside formal parameter, no changes will be observed inside actual parameter.

So, after execution of sub-program, values of actual parameter will remain same.

list [2] = { 1, 3 };

(b) Passed by reference.
~~Pa~~ Pass by reference means to pass reference of an argument to formal parameters.
So any changes made in formal parameter will be reflected in actual parameters.

After execution of sub program and function fun ?,
list [2] = { 2, 6 }.

(c) Passed by value-result.
When 'passed by value-result' is executed, values of actual parameters are copied back to formal parameter. When execution ends, values of formal parameters are ~~again~~ copied to actual parameters

DATE
PAGE

After ~~stac~~ execution of sub-program and ~~function (fun).~~ function (fun).

list[2] = {2, 6}