Parameter Passing Example from a Previous Exam

void main()

{

int aM[] = {10, 20, 30}

int k = 2;

sub(aM[k], k);

print ('main', k, aM[0], aM[1], aM[2]);

void sub(int x, int y)

{

y -= 1;

x += 3;

k -= 1;

printf('sub', x, y);

}

}

Trace and show the output produced by the code using the following parameter transmission schemes:

a. by value

b. by true reference (change actual value)

c. by copy-restore reference (after subroutine replace beginning values/parameters with results)

d. by name (resolve in real timeish)

|  |  |  |
| --- | --- | --- |
|  | **Problem a. by value** | **Problem b. by true ref** |
| main | aM 10 20 30  k 2 1 | aM 10 20 30  33  k 2 1 0 |
| sub | x 30 33  y 2 1 | x @aM[2] , @aM[2]=33  y @k |
| Output: | Sub 33 1  main 1 10 20 30 | Sub 33 0  Main 0 10 20 33 |

|  |  |  |
| --- | --- | --- |
|  | **Problem c. by copy-restore** | **Problem d. by name** |
| main | aM 10 20 30  33  k 2 1 1 | aM 10 20 30  23  k 2 1 0 |
| sub | X 30 33  Y 2 1 | x @aM[k]  y @k |
| Output: | Sub 33 1  main 1 10 20 33 | Sub 10 0  Main 0 10 23 30 |

Parameter Passing Example from a Previous Exam

void main()

{

int aM[] = {10, 20, 30}

int k = 2;

sub(aM[k], k);

print ('main', k, aM[0], aM[1], aM[2]);

void sub(int x, int y)

{

y -= 1;

x += 3;

k -= 1;

printf('sub', x, y);

}

}

Trace and show the output produced by the code using the following parameter transmission schemes:

a. by value

b. by true reference

c. by copy-restore reference

d. by name

|  |  |  |
| --- | --- | --- |
|  | **Problem a. by value** | **Problem b. by true ref** |
| main | aM 10 20 30  k 2 1 | aM 10 20 30  33  k 2 1 0 |
| sub | X 30 33  Y 2 1 | X @aM[2]  Y @k |
| Output: | Sub 33 1  main 1 10 20 30 | Sub 33 0  Main 0 10 20 33 |

|  |  |  |
| --- | --- | --- |
|  | **Problem c. by copy-restore** | **Problem d. by name** |
| main | aM 10 20 30  33  k 2 1 1 | aM 10 20 30  23  k 2 1 0 |
| sub | X 30 33  Y 2 1 | X @aM[k]  Y @k |
| Output: | Sub 33 1  main 1 10 20 33 | Sub 10 0  Main 0 10 23 30  **//EACH IN REAL TIME** |