Batch:G3   Roll No.: 16010421063

Experiment / assignment / tutorial No. 8

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

TITLE: Virtual Lab experiment on Call by reference.

AIM: Virtual Lab experiment on Call by reference

http://cse02-iiith.vlabs.ac.in/

http://cse02-iiith.vlabs.ac.in/exp8/simulation/CallByReferencePointers/index.html

Program to swap two number without using third variable using Call by reference.

Expected OUTCOME of Experiment:

CO4: Design modular programs using functions and demonstrate the concept of pointers and file handling

Books/ Journals/ Websites referred:

1. Programming in C, second edition, Pradeep Dey and Manas Ghosh, Oxford University Press.
2. Programming in ANSI C, fifth edition, E Balagurusamy, Tata McGraw Hill.
3. Introduction to programming and problem solving , G. Michael Schneider ,Wiley India edition.
4. **http://cse.iitkgp.ac.in/~rkumar/pds-vlab/**

Problem Definition:

The **call by reference** method of passing arguments to a function copies the address of an argument into the formal parameter. Inside the function, the address is used to access the actual argument used in the call.

In the program, a function called swap ( ) is used in which the address is used to access the actual argument.

Algorithm:

Step1: Start

Step2: Declare variable A and B as Integer data type and assign value 5 and 9 respectively.

Step 3: Print A

Step 4: Print B

Step 5: Call function swap() and pass A and B as agreements

Step 6: Function call. Local variable type pointer Pa and Pb get assigned the address of A and B respectively.

Step 7: A local variable is declared temp and assigned the value at Address Pa

Step 8: Variable Pa is assigned the value at Address Pb

Step 9: Variable Pb is assigned the value at temp

Step 10: Print A

Step 11: Print B

Step 12: Stop

Implementation details:

```c
#include<stdio.h>
 void main(){
int A = 5, B = 9;
printf("Value of A is %d\n",A);
printf("Value of B is %d\n",B);
swap( &A , &B );
printf("Value of A after swapping is %d\n",A);
printf("Value of B after swapping is %d\n",B);
 }

 void swap( int *Pa , int *Pb)
 {
int temp = *Pa;
*Pa = *Pb;
*Pb = temp;
 }
```

Output(s):

(Attach screenshots of output of the Program Code implemented in Virtual Lab and Quiz attempted)

K. J. Somaiya College of Engineering, Mumbai-77

(A Constituent College of Somaiya Vidyavihar University)

Pointers

Program Code

```
#include<stdio.h>
void main(){
    int A = 5, B = 9;
    printf("Value of A is %d\n",A);
    printf("Value of B is %d\n",B);
    swap( &A , &B );
    printf("Value of A after swapping is %d\n",A);
    printf("Value of B after swapping is %d\n",B);
}

void swap( int *Pa , int *Pb){
    int temp = *Pa;
    *Pa = *Pb;
```

Back Next

Memory Map

Address	BYTE 1	BYTE 2	BYTE 3	BYTE 4	Variable
60	0	0	0	5	A
56	0	0	0	9	B
52					
48					
44					
40					
36					
32					
28					
24					
20					
16					
12					
8			Program Memory		
4			Program Memory		
0			Reserved By Os		

Code Output

Explanation

Variables A and B are declared as integer types and assigned values 5 and 9 respectively.

Pointers

Program Code

```
#include<stdio.h>
void main(){
    int A = 5, B = 9;
    printf("Value of A is %d\n",A);
    printf("Value of B is %d\n",B);
    swap( &A , &B );
    printf("Value of A after swapping is %d\n",A);
    printf("Value of B after swapping is %d\n",B);
}

void swap( int *Pa , int *Pb){
    int temp = *Pa;
    *Pa = *Pb;
```

Back Next

Memory Map

Address	BYTE 1	BYTE 2	BYTE 3	BYTE 4	Variable
60	0	0	0	9	A
56	0	0	0	5	B
52					
48					
44					
40	0	0	0	60	Pa
36	0	0	0	56	Pb
32					
28					
24					
20					
16					
12					
8			Program Memory		
4			Program Memory		
0			Reserved By Os		

Code Output

Explanation

Function call. Local integer type pointer variables Pa and Pb get assigned the address of A and B respectively.

Pointers

Program Code

```
void main(){
    int A = 5, B = 9;
    printf("Value of A is %d\n",A);
    printf("Value of B is %d\n",B);
    swap( &A , &B );
    printf("Value of A after swapping is %d\n",A);
    printf("Value of B after swapping is %d\n",B);
}

void swap( int *Pa , int *Pb){
    int temp = *Pa;
    *Pa = *Pb;
    *Pb = temp;
}
```

Back Next

Memory Map

Address	BYTE 1	BYTE 2	BYTE 3	BYTE 4	Variable
60	0	0	0	9	A
56	0	0	0	5	B
52					
48					
44					
40	0	0	0	60	Pa
36	0	0	0	56	Pb
32	0	0	0	5	temp
28					
24					
20					
16					
12					
8			Program Memory		
4			Program Memory		
0			Reserved By Os		

Code Output

Explanation

A local integer type variable 'temp' is declared and the value at address in Pa is assigned to it i.e. A's value is stored in temp.

Department of Science and Humanities

K. J. Somaiya College of Engineering, Mumbai-77

(A Constituent College of Somaiya Vidyavihar University)

Pointers

Program Code

```
#include<stdio.h>
void main(){
    int A = 5, B = 9;
    printf("Value of A is %d\n",A);
    printf("Value of B is %d\n",B);
    swap( &A , &B );
    printf("Value of A after swapping is %d\n",A);
    printf("Value of B after swapping is %d\n",B);
}

void swap( int *Pa , int *Pb){
    int temp = *Pa;
    *Pa = *Pb;
```

Back   Next

Memory Map

Address	BYTE 1	BYTE 2	BYTE 3	BYTE 4	Variable
60	0	0	0	9	A
56	0	0	0	5	B
52					
48					
44					
40					
36					
32					
28					
24					
20					
16					
12					
8		Program Memory			
4		Program Memory			
0		Reserved By Os			

Code Output

Value of A after swapping is 9

Explanation

The New Value stored in variable A is 9 which is displayed in the output.

Pointers

Program Code

```
#include<stdio.h>
void main(){
    int A = 5, B = 9;
    printf("Value of A is %d\n",A);
    printf("Value of B is %d\n",B);
    swap( &A , &B );
    printf("Value of A after swapping is %d\n",A);
    printf("Value of B after swapping is %d\n",B);
}

void swap( int *Pa , int *Pb){
    int temp = *Pa;
    *Pa = *Pb;
```

Back   Next

Memory Map

Address	BYTE 1	BYTE 2	BYTE 3	BYTE 4	Variable
60	0	0	0	9	A
56	0	0	0	5	B
52					
48					
44					
40					
36					
32					
28					
24					
20					
16					
12					
8		Program Memory			
4		Program Memory			
0		Reserved By Os			

Code Output

Value of B after swapping is 5

Explanation

The New Value stored in variable B is 5 which is displayed in the output.

Pointers

Program Code

```
#include<stdio.h>
void main(){
    int A = 5, B = 9;
    printf("Value of A is %d\n",A);
    printf("Value of B is %d\n",B);
    swap( &A , &B );
    printf("Value of A after swapping is %d\n",A);
    printf("Value of B after swapping is %d\n",B);
}

void swap( int *Pa , int *Pb){
    int temp = *Pa;
    *Pa = *Pb;
```

Back   Next

Memory Map

Address	BYTE 1	BYTE 2	BYTE 3	BYTE 4	Variable
60	0	0	0	9	A
56	0	0	0	5	B
52					
48					
44					
40					
36					
32					
28					
24					
20					
16					
12					
8		Program Memory			
4		Program Memory			
0		Reserved By Os			

Code Output

Explanation

Program Execution Complete

Pointers

1. What would be the equivalent pointer expression for referring the array element a[i][j][k][l]

- a: ((((a+i)+j)+k)+l)
- ● b: *(*(*(*(a+i)+j)+k)+l)
- c: (((a+i)+j)+k+l)
- d: ((a+i)+j+k+l)

2. NULL pointer points to the 0th memory address:

- ● a: True
- b: False

Submit Quiz

Department of Science and Humanities

Pointers

1. Pointer is a :
◉ a: A keyword used to create variables
○ b: A variable that stores address of an instruction
○ c: A variable that stores address of other variable
○ d: All of the above

2. If a variable is a pointer to a structure, then which of the following operator is used to access data members of the structure through the pointer variable?:
○ a: .
○ b: &
○ c: *
◉ d: ->

3.The name of the array is a pointer to the _____ element of the array.
◉ a: first
○ b: second

[Submit Quiz]

3 out of 3

Conclusion:

We learnt the execution of the call by reference program. We were also able to design modular programs using functions and demonstrate the concept of pointers

Post Lab Descriptive Questions

1. Difference between call by value and call by reference in C.

Call By Value	Call By Reference
While calling a function, we pass values of variables to it. Such functions are known as Call By Values.	While calling a function, we pass address of variables to the function known as "Call By References
In this method, the value of each variable in calling function is copied into corresponding dummy variables of the called function.	In this method, the address of actual variables in the calling function are copied into the dummy variables of the called function.
Changes made in a copy of variable never modify the value of variable outside the function	Change in the variable also affects the value of the variable outside the function
In call by values we cannot alter the values of actual variables through function calls.	In call by reference we can alter the values of variables through function calls.

2. **What would be the output of the following program:**

```
main( )
{
        float a = 13.5 ;
        float *b, *c ;
        b = &a ; /* suppose address of a is 1006 */
        c = b ;
        printf ( "\n%u %u %u", &a, b, c ) ;
        printf ( "\n%f %f %f %f %f", a, *(&a), *&a, *b, *c ) ;
}
```

Output is:
775117564 775117564 775117564
13.500000 13.500000 13.500000 13.500000 13.500000

(This output was generated using Codechef online IDE.)

Date: **Signature of faculty**
in-charge