Higher School of Applied Sciences of Tlemcen
1-year computer science course

Functions

Part 2

Presented by:
Dr. Imane NEDJAR

Part 2

Passing by Reference and Passing by Value

Functions

Passing by Reference and Passing by Value

Problem Statement

```
#include<stdio.h>
int x,y;
void permutation (int a,int b);
void permutation (int a,int b){
int c;
c=a;
a=b;
b=c;}
int main(){
printf("Provide the value of x:");
scanf("%d",&x);
printf(""Provide the value of y:");
scanf("%d",&y);
permutation(x,y);
printf("x = %d\n",x);
printf("y = %d\n",y);
return 0;}
```

@7	3	x
@6	2	y
@5		
@4		
@3		a
@2		b
@1		c
@0		

@7	3	x
@6	2	y
@5		
@4		
@3	3	a
@2	2	b
@1		c
@0		

@7	3	x
@6	2	y
@5		
@4		
@3	2	a
@2	3	b
@1	3	c
@0		

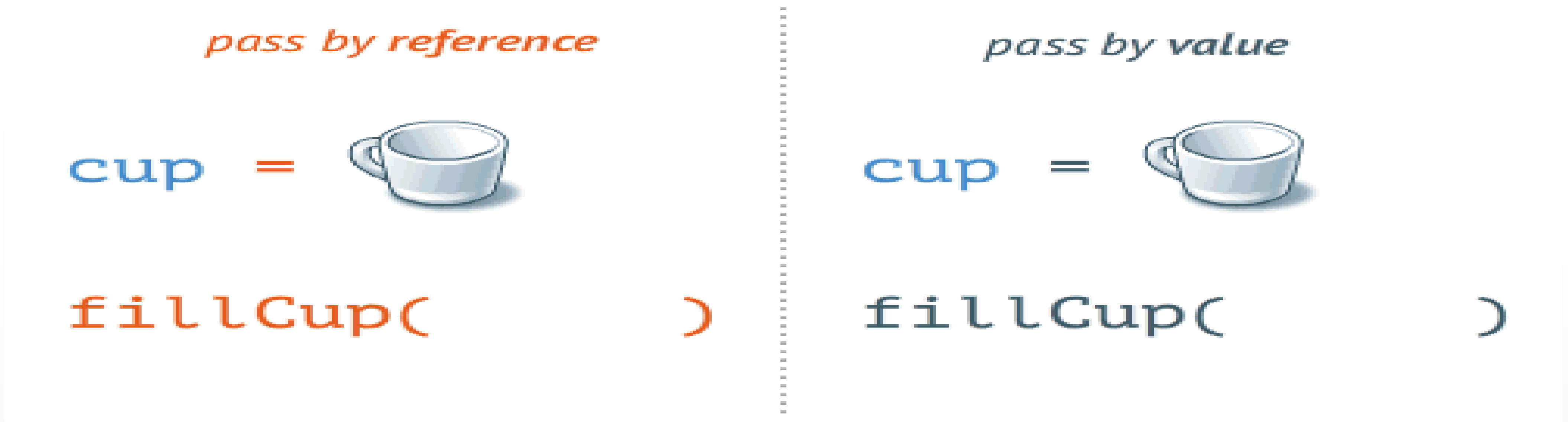
```
Provide the value of x:3
Provide the value of y:2
x=3
y=2
```

Passing by Reference and Passing by Value

pass by reference
cup =
fillCup( )
pass by value
cup =
fillCup( )
www.penjee.com

Functions

Passing by Reference and Passing by Value



In pass by address, you pass the address of the original argument. **The changes made are permanent**

However, in pass by value, a copy of the original argument is manipulated, which **does not affect the original argument**

Passing by Reference and Passing by Value

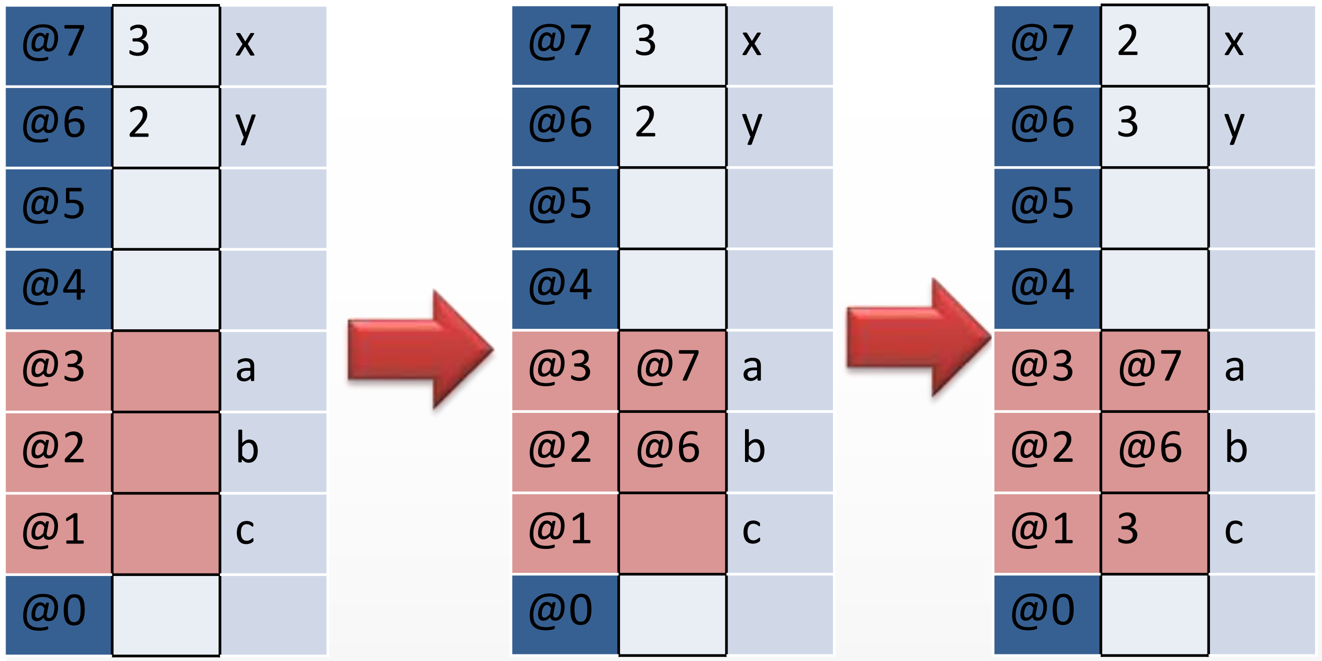
```
#include<stdio.h>
int x,y;
void permutation (int a,int b);
void permutation (int a,int b){
int c;
c=a;
a=b;
b=c;}
int main(){
printf("Provide the value of x:");
scanf("%d",&x);
printf(""Provide the value of y:");
scanf("%d",&y);
permutation(x,y);
printf("x = %d\n",x);
printf("y = %d\n",y);
return 0;}
```

```
#include<stdio.h>
int x,y;
void permutation (int*a, int*b);
void permutation (int*a, int*b){
int c;
c=*a;
*a=*b;
*b=c;}
int main(){
printf("Provide the value of x:");
scanf("%d",&x);
printf(""Provide the value of y:");
scanf("%d",&y);
permutation(&x,&y);
printf("x = %d\n",x);
printf("y = %d\n",y);
return 0;}
```

Functions

Passing by Reference and Passing by Value

```
#include<stdio.h>
int x,y;
void permutation (int*a, int*b);
void permutation (int*a, int*b){
int c;
c=*a;
*a=*b;
*b=c;}
int main(){
printf("Provide the value of x:");
scanf("%d",&x);
printf(""Provide the value of y:");
scanf("%d",&y);
permutation(&x,&y);
printf("x = %d\n",x);
printf("y = %d\n",y);
return 0;}
```



@7	3	x		@7	3	x		@7	2	x
@6	2	y		@6	2	y		@6	3	y
@5				@5				@5		
@4				@4				@4		
@3		a		@3	@7	a		@3	@7	a
@2		b		@2	@6	b		@2	@6	b
@1		c		@1		c		@1	3	c
@0				@0				@0		

```
Provide the value of x:3
Provide the value of y:2
x=2
y=3
```

Functions

Passing by Reference and Passing by Value

```
#include<stdio.h>
int x,y;
void permutation (int a,int b);
void permutation (int a,int b){
int c;
c=a;
a=b;
b=c;}
int main(){
printf("Provide the value of x:");
scanf("%d",&x);
printf(""Provide the value of y:");
scanf("%d",&y);
permutation(x,y);
printf("x = %d\n",x);
printf("y = %d\n",y);
return 0;}
```

The **actual parameters x and y** pass their values to the **formal parameters a and b** => **pass by value**

- **Passing by value does not allow for changing the values of the actual parameters.**

Functions

Passing by Reference and Passing by Value

```
#include<stdio.h>
int x,y;
void permutation (int*a, int*b);
void permutation (int*a, int*b){
int c;
c=*a;
*a=*b;
*b=c;}
int main(){
printf("Provide the value of x:");
scanf("%d",&x);
printf(""Provide the value of y:");
scanf("%d",&y);
permutation(&x,&y);
printf("x = %d\n",x);
printf("y = %d\n",y);
return 0;}
```

- The formal parameters a et b **must contain addresses** and not values

- **Passing the addresses of x and y to the permutation function in order to modify this memory location.**

Passing by Reference and Passing by Value

```
scanf("%d",&y);
```

Functions

Passing by Reference and Passing by Value

Arrays

```
#include<stdio.h>
void Read_Array (int t[], int N){
int i;
for(i=0;i<N;i++)
{printf("t[%d]=",i);
scanf("%d",&t[i]);
   } }
```

```
int main(){
    int t[20],N,i;
printf("Provide the size of the array:");
scanf("%d",&N);
Read_Array(t,N);
for(i=0;i<N;i++)
printf("t[%d]=%d\n",i,t[i]);
return 0;}
```

Functions

Passing by Reference and Passing by Value

String

```
#include<stdio.h>
#include<string.h>
void Modify_text (char txt []){
    int i;
    for (i=0;i<strlen(txt);i++)
    txt[i]='i';
        }
```

```
int main(){
    char txt[30];
    printf("Provide the  text\n");
    gets(txt);
    Modify_text (txt);
    printf("%s",txt);
    return 0;}
```

Passing by Reference and Passing by Value

String

strcat (char s1[], **char** s2[]**);**

- To append **S2** at the end of **S1**

strcpy (char s1[], **char** s2[]**);**

- Assigns **S1** to **S2**

Functions

Exercice 1

Let's consider the increment procedure that increments a variable's value by 4. Complete the following program

```
#include<stdio.h>
int x;
void  incrementation (?){
?=?+4;
}
int main(){
    printf("Entrer la valeur de x :");
    scanf("%d",&x);
    incrementation (?);
    printf("x = %d",?);
return 0;}
```

Functions

Exercice 2

Create a procedure that takes two strings as input and returns their length.

Exercice 3

Create a procedure that takes two integers as input and computes and returns their factorial