Part 4 (FUNCTION)

إجمالي النقاط 22/34

عنوان بريد إلكتروني *

abdallah.shabaan.ghazy@gmail.com

22 من إجمالي 34 نقطة

0/1

* Q25)Point out the error in the program

```c
#include<stdio.h>

int main()
{
    int a=10;
    void f();
    a = f();
    printf("%d\n", a);
    return 0;
}
void f()
{
    printf("Hi");
}
```

- ○ A. Error: Not allowed assignment
- ○ B. Error: Doesn't print anything
- ○ C. No error
- ◉ D. None of above

الإجابة الصحيحة

- ◉ A. Error: Not allowed assignment

التعليقات

The function void f() is not visible to the compiler while going through main() function. So we have to declare this prototype void f(); before to main() function. This kind of error will not occur in modern compilers.

0/1 * ? Q35) Can A program be compiled without a main function ✗

- ○ yes
- ◉ No

الإجابة الصحيحة

- ◉ yes

1/1

* ?Q13)What will be the output of the C program ✓

```c
#include<stdio.h>
int sumdig(int);
int main()
{
    int a, b;
    a = sumdig(123);
    b = sumdig(123);
    printf("%d, %d\n", a, b);
    return 0;
}
int sumdig(int n)
{
    int s, d;
    if(n!=0)
    {
        d = n%10;
        n = n/10;
        s = d+sumdig(n);
    }
    else
        return 0;
    return s;
}
```

- ○ A. 4, 4
- ○ B. 3, 3
- ◉ C. 6, 6
- ○ D. 12, 12

0/1

* ?Q19)What will be the output of the program

```c
#include<stdio.h>

int fun(int i)
{
    i++;
    return i;
}

int main()
{
    int fun(int);
    int i=3;
    fun(i=fun(fun(i)));
    printf("%d\n", i);
    return 0;
}
```

- A. 5
- B. 4
- C. Error
- D. Garbage value
- E. 6

الإجابة الصحيحة

A. 5

التعليقات

Step 1: int fun(int); This is prototype of function fun(). It tells the compiler that the function .fun() accept one integer parameter and returns an integer value

.Step 2: int i=3; The variable i is declared as an integer type and initialized to value 3

Step 3: fun(i=fun(fun(i)));. The function fun(i) increements the value of i by 1(one) and .return it

,Lets go step by step

.fun(i) becomes fun(3) is called and it returns 4 <=

i = fun(fun(i)) becomes i = fun(4) is called and it returns 5 and stored in variable i.(i=5) <=

fun(i=fun(fun(i))); becomes fun(5); is called and it return 6 and nowhere the return value <= .is stored

Step 4: printf("%d\n", i); It prints the value of variable i.(5)

.'Hence the output is '5

0/1

* ?Q14)What will be the output of the C program

```c
#include<stdio.h>

int main()
{
    int fun(int);
    int i = fun(10);
    printf("%d\n", --i);
    return 0;
}
int fun(int i)
{
   return (i++);
}
```

- A. 9
- B. 10
- C. 11
- D. 8

الإجابة الصحيحة

A. 9

التعليقات

.()Step 1: int fun(int); Here we declare the prototype of the function fun

Step 2: int i = fun(10); The variable i is declared as an integer type and the result of the .fun(10) will be stored in the variable i

Step 3: int fun(int i){ return (i++); } Inside the fun() we are returning a value return(i++). It .returns 10. because i++ is the post-increement operator

Step 4: Then the control back to the main function and the value 10 is assigned to variable .i

.Step 5: printf("%d\n", --i); Here --i denoted pre-increement. Hence it prints the value 9

1/1

* ?Q9)What will be the output of the program ✓

```
#include<stdio.h>
void fun(int);
typedef int (*pf) (int, int);
int proc(pf, int, int);

int main()
{
    int a=3;
    fun(a);
    return 0;
}
void fun(int n)
{
    if(n > 0)
    {
        fun(--n);
        printf("%d,", n);
        fun(--n);
    }
}
```

- ○ A. 0, 2, 1, 0,
- ○ B. 1, 1, 2, 0,
- ○ C. 0, 1, 0, 2,
- ◉ D. 0, 1, 2, 0,

1/1

* ?Q20)What will be the output of the program ✓

```c
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int i=0;
    i++;
    if(i<=5)
    {
        printf("IndiaBIX");
        exit(1);
        main();
    }
    return 0;
}
```

- ○ A. Prints "IndiaBIX" 5 times
- ○ B. Function main() doesn't calls itself
- ○ C. Infinite loop
- ◉ D. Prints "IndiaBIx"

1/1 * ?Q26)Which of the following statements are correct about the program ✓

```c
#include<stdio.h>

int main()
{
    printf("%p\n", main());
    return 0;
}
```

- ○ A. It prints garbage values infinitely
- ◉ B. Runs infinitely without printing anything
- ○ C. Error: main() cannot be called inside printf()
- ○ D. No Error and print nothing

0/1* Q3)In C, what is the meaning of following function prototype with empty parameter list ✕

```
void fun()
{
   /* .... */
}
```

- (A) Function can only be called without any parameter
- (B) Function can be called with any number of parameters of any types
- (C) Function can be called with any number of integer parameters.
- (D) Function can be called with one integer parameter.

الإجابة الصحيحة

- (B) Function can be called with any number of parameters of any types

التعليقات

Empty list in C mean that the parameter list is not specified and function can be called with any parameters. In C, to declare a function that can only be called without any "parameter, we should use "void fun(void)

1/1

* ?Q7)What will be the output of the program ✓

```c
#include<stdio.h>
int i;
int fun();

int main()
{
    while(i)
    {
        fun();
        main();
    }
    printf("Hello\n");
    return 0;
}
int fun()
{
    printf("Hi");
}
```

- (●) A. Hello
- ( ) B. Hi Hello
- ( ) C. No output
- ( ) D. Infinite loop

0/1

* ?Q12)What will be the output of the C program

```
#include<stdio.h>
int function();
main()
{
        int i;
        i = function();
        printf("%d", i);
        return 0;
}
function()
{
        int a;
        a = 250;
}
```

- A.250
- B. 0
- C. 1
- D. Some
- Garbage value

الإجابة الصحيحة

C. 1

التعليقات

Here function executed successfully and we don't return anything, by default C compiler returns 1 for its successful execution.

1/1 * ?Q16)What will be the output of the program ✓

```
#include<stdio.h>

int addmult(int ii, int jj)
{
    int kk, ll;
    kk = ii + jj;
    ll = ii * jj;
    return (kk, ll);
}

int main()
{
    int i=3, j=4, k, l;
    k = addmult(i, j);
    l = addmult(i, j);
    printf("%d %d\n", k, l);
    return 0;
}
```

- (●) [A].12 12
- ( ) [B]. No error, No output
- ( ) [C]. Error: Compile error
- ( ) [D]. None of above

1/1* Q30) A function may have any number of return statements each returning different values ✓

- (●) True
- ( ) False

1/1 * ?Q18)What will be the output of the program ✓

```c
#include<stdio.h>
int func1(int);

int main()
{
    int k=35;
    k = func1(k=func1(k=func1(k)));
    printf("k=%d\n", k);
    return 0;
}
int func1(int k)
{
    k++;
    return k;
}
```

- ○ A. k=35
- ○ B. k=36
- ○ C. k=37
- ◉ D. k=38

1/1* Q5)The keyword used to transfer control from a function back to the .....calling function is ✓

- ○ A.switch
- ○ B. goto
- ○ C. go back
- ◉ D. return

void main() {… ; main();} (what happen)

- ○ Compilation error
- ○ Linking error
- ◉ stack overflow (Recursion function)

1/1

* ?Q15)What will be the output of the program ✓

```c
#include<stdio.h>

int main()
{
    int i=1;
    if(!i)
        printf("IndiaBIX,");
    else
    {
        i=0;
        printf("C-Program");
        main();
    }
    return 0;
}
```

- ○ A. prints "IndiaBIX, C-Program" infinitely
- ◉ B. prints "C-Program" infinetly
- ○ C. prints "C-Program, IndiaBIX" infinitely
- ○ D. Error: main() should not inside else statement

1/1

* Q24)Point out the error in the program ✓

```
#include<stdio.h>
int f(int a)
{
  a > 20? return(10): return(20);
}
int main()
{
    int f(int);
    int b;
    b = f(20);
    printf("%d\n", b);
    return 0;
}
```

- ○ A. Error: Prototype declaration
- ○ B. No error
- ◉ C. Error: return statement cannot be used with conditional operators
- ○ D. None of above

0/1

* .Q32)Usually recursion works slower than loops ✗

- ○ Yes
- ◉ No

الإجابة الصحيحة

- ◉ Yes

التعليقات

When a recursive call is made, the function/process clones itself and then process that .funtion. This leads to time and space constrains

.In a loop, there is no recursive call involved that saves a lot of time and space too

1/1

* Q29)Functions can be called either by value or reference ✓

- ◉ True
- ○ False

0/1 * ?Q33)Which of the following is true about return type of functions in C

- (A) Functions can return any type
- (B) Functions can return any type except array and functions
- (C) Functions can return any type except array, functions and union
- (D) Functions can return any type except array, functions, function pointer and union

الإجابة الصحيحة

(B) Functions can return any type except array and functions

التعليقات

Explanation: In C, functions can return any type except arrays and functions. We can get around this limitation by returning pointer to array or pointer to function.

0/1 * Q1) A function cannot be defined inside another function

- TRUE
- FALSE

الإجابة الصحيحة

TRUE

التعليقات

A function cannot be defined inside the another function, but a function can be called inside a another function.

1/1

* (Q27 ✓

Which of the following statements are correct about the function?

```
long fun(int num)
{
    int i;
    long f=1;
    for(i=1; i<=num; i++)
        f = f * i;
    return f;
}
```

- ○ A. The function calculates the value of 1 raised to power num.
- ○ B. The function calculates the square root of an integer
- ◉ C. The function calculates the factorial value of an integer
- ○ D. None of above

0/1* Q2)What will be the output of the program If characters 'a', 'b' and 'c' ?enter are supplied as input

```c
#include<stdio.h>

int main()
{
    void fun();
    fun();
    printf("\n");
    return 0;
}
void fun()
{
    char c;
    if((c = getchar())!= '\n')
        fun();
    printf("%c", c);
}
```

- A. abc abc
- B. bca
- C. Infinite loop
- D. cba

الإجابة الصحيحة

D. cba

التعليقات

.()Step 1: void fun(); This is the prototype for the function fun

.Step 2: fun(); The function fun() is called here

The function fun() gets a character input and the input is terminated by an enter key(New .line character). It prints the given character in the reverse order

"The given input characters are "abc

Output: cba

1/1 * Q28)Functions cannot return more than one value at a time ✓

- (●) True
- ( ) False

1/1

* ?Q17)What will be the output of the program ✓

```
#include<stdio.h>
int i;
int fun1(int);
int fun2(int);

int main()
{
    extern int j;
    int i=3;
    fun1(i);
    printf("%d,", i);
    fun2(i);
    printf("%d", i);
    return 0;
}
int fun1(int j)
{
    printf("%d,", ++j);
    return 0;
}
int fun2(int i)
{
    printf("%d,", ++i);
    return 0;
}
int j=1;
```

- ○ A. 3, 4, 4, 3
- ◉ B. 4, 3, 4, 3
- ○ C. 3, 3, 4, 4
- ○ D. 3, 4, 3, 4

1/1

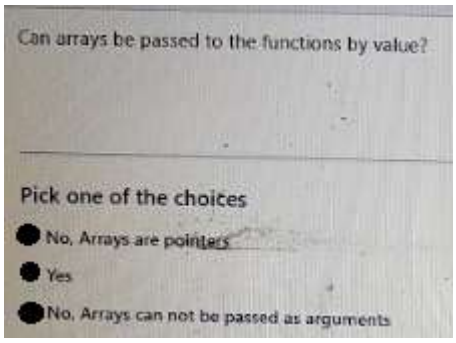
* ?Q34)Output of following program ✓

```c
#include<stdio.h>

void dynamic(int s, ...)
{
    printf("%d ", s);
}

int main()
{
    dynamic(2, 4, 6, 8);
    dynamic(3, 6, 9);
    return 0;
}
```

- (A) 2 3
- (B) Compiler Error
- (C) 4 3
- (D) 3 2

1/1

* (Q22 ✓



- Option 1
- Option 2
- Option 3

0/1

* ?Q8)What will be the output of the program

```
#include<stdio.h>
int reverse(int);

int main()
{
    int no=5;
    reverse(no);
    return 0;
}
int reverse(int no)
{
    if(no == 0)
        return 0;
    else
        printf("%d,", no);
    reverse (no--);
}
```

- A. Print 5, 4, 3, 2, 1
- B. Print 1, 2, 3, 4, 5
- C. Print 5, 4, 3, 2, 1, 0
- D. Infinite loop

الإجابة الصحيحة

D. Infinite loop

التعليقات

.Step 1: int no=5; The variable no is declared as integer type and initialized to 5

Step 2: reverse(no); becomes reverse(5); It calls the function reverse() with '5' as .parameter

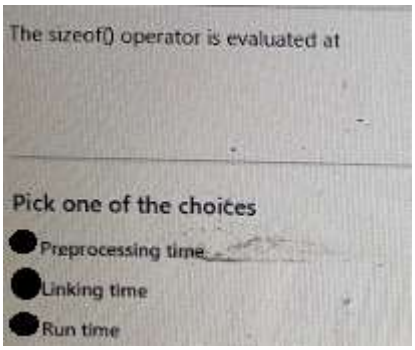
The function reverse accept an integer number 5 and it returns '0'(zero) if(5 == 0) if the given number is '0'(zero) or else printf("%d,", no); it prints that number 5 and calls the .;function reverse(5)

The function runs infinetely because the there is a post-decrement operator is used. It will not decrease the value of 'n' before calling the reverse() function. So, it calls reverse(5) .infinitely

Note: If we use pre-decrement operator like reverse(--n), then the output will be 5, 4, 3, 2, 1.
'Because before calling the function, it decrements the value of 'n

1/1

* (Q4 ✓



- ○ Option 1
- ○ Option 2
- ◉ Option 3

0/1

* Q21)Point out the error in the program

```
f(int a, int b)
{
    int a;
    a = 20;
    return a;
}
```

- ○ A. Missing parenthesis in return statement
- ◉ B. The function should be defined as int f(int a, int b)
- ○ C. Redeclaration of a
- ○ D. None of above

الإجابة الصحيحة

◉ C. Redeclaration of a

التعليقات

.f(int a, int b) The variable a is declared in the function argument statement

int a; Here again we are declaring the variable a. Hence it shows the error "Redeclaration of "a

1/1 * ? "Q6)How many times the program will print "IndiaBIX ✓

```c
#include<stdio.h>

int main()
{
    printf("IndiaBIX");
    main();
    return 0;
}
```

- ○ A. Infinite times
- ○ B. 32767 times
- ○ C. 65535 times
- ◉ D. Till stack overflows

1/1* Q31)Names of functions in two different files linked together must be unique ✓

- ◉ True
- ○ False

1/1 * Q10)If return type for a function is not specified, it defaults to int ✓

- ◉ true
- ○ false

1/1

* ?Q11)What will be the output of the C program ✓

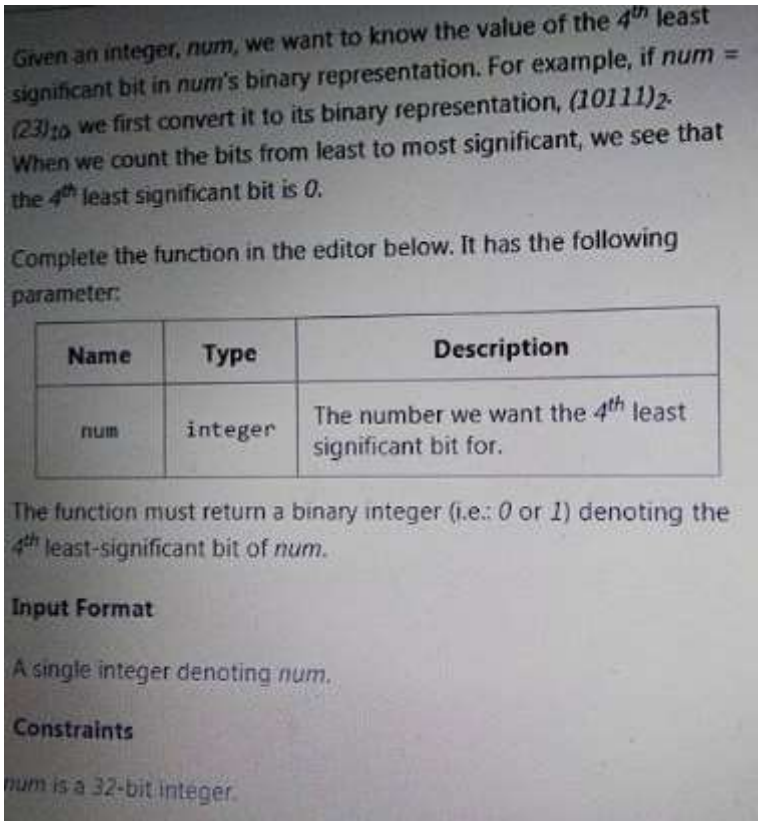
```
#include<stdio.h>
int main()
{
        function();
        return  0;
}
void function()
{
        printf("Function in C is awesome");
}
```

- ○ A. Function in C is awesome
- ○ B. no output
- ○ C. Runtime error
- ◉ D. Compilation error

0 من إجمالي 0 نقطة

write c program

* (Q6



Given an integer, *num*, we want to know the value of the 4^{th} least significant bit in *num*'s binary representation. For example, if *num* = $(23)_{10}$ we first convert it to its binary representation, $(10111)_2$. When we count the bits from least to most significant, we see that the 4^{th} least significant bit is *0*.

Complete the function in the editor below. It has the following parameter:

Name	Type	Description
num	integer	The number we want the 4^{th} least significant bit for.

The function must return a binary integer (i.e.: *0* or *1*) denoting the 4^{th} least-significant bit of *num*.

Input Format

A single integer denoting *num*.

Constraints

num is a 32-bit integer.

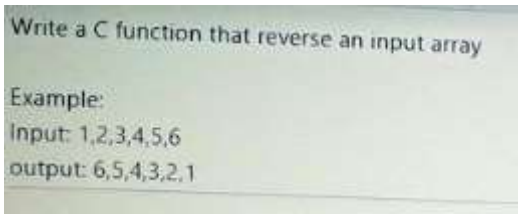
-

التعليقات

```
()int main
}
;int n, c, k
;printf("Enter an integer in decimal number system\n")
;fflush(stdin);fflush(stdout)
;scanf("%d", &n)

;printf("%d in binary number system is:\n", n)

for (c = 31; c >= 0; c--)
}
;k = n >> c

if (k & 1)
;printf("1")
else
;printf("0")
{

;printf("\n")
;k=n>>3
if(k&1)
```

```
    printf(" 4th least significant bit is 1");
else
    printf("4th least significant bit is 0");
    return 0;
{
```

* (Q3



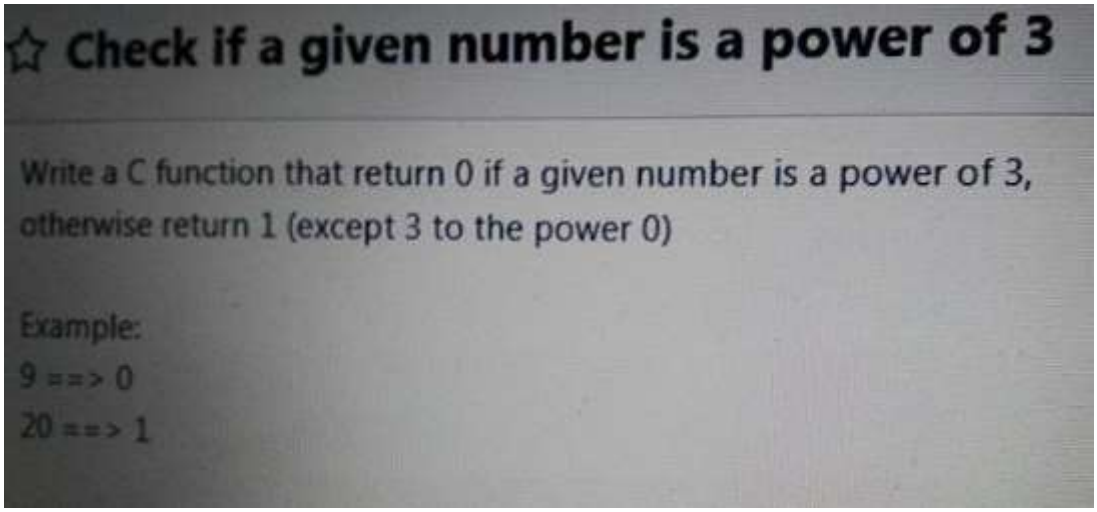
-

التعليقات

```
;void inp_array(int arr[],int size)
;void reverse_array(int arr[],int size)
;void print_array(int arr[],int size)
()int main
}
;int arr[10]
;int size

;printf("enter size of array")
;fflush(stdin);fflush(stdout)
;scanf("%d",&size)

;printf("enter %d element of array",size)
;fflush(stdin);fflush(stdout)
;inp_array(arr,size)

;printf("element before reverse\n")
;print_array(arr,size)

;reverse_array(arr,size)

;printf("\nelement after reverse \n")
;print_array(arr,size)

{
void inp_array(int arr[],int size)
}
;int i
for(i=0;i<size;i++)
}
;scanf("%d",&arr[i])
{
{
void print_array(int arr[],int size)
}
;int i

for(i=0;i<size;i++)
}
;printf(" %d",arr[i])
```

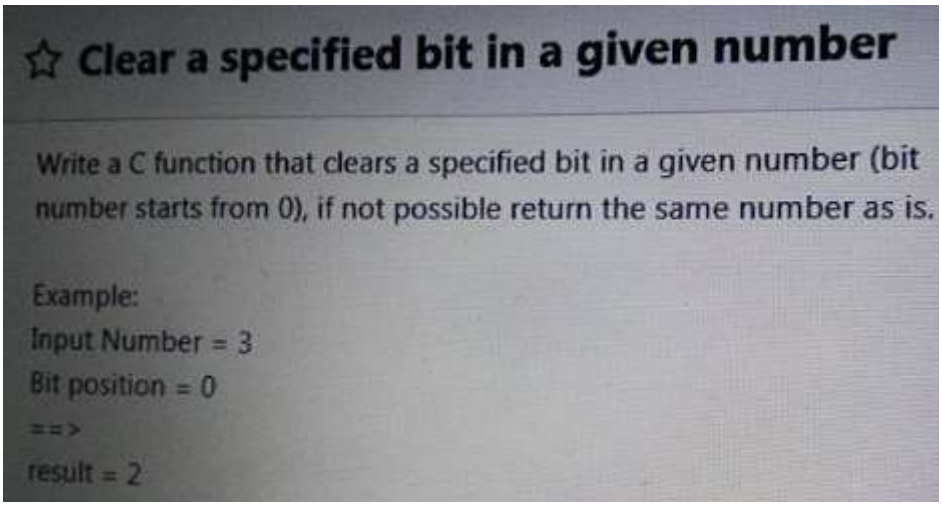
```
{
{
void reverse_array(int arr[],int size)
}
;int i,j,temp
for(i=0,j=size-1;i<=j;i++,j--)
}
;temp=arr[i]
;arr[i]=arr[j]
;arr[j]=temp

{

{
```

* Q7



-

* (Q5

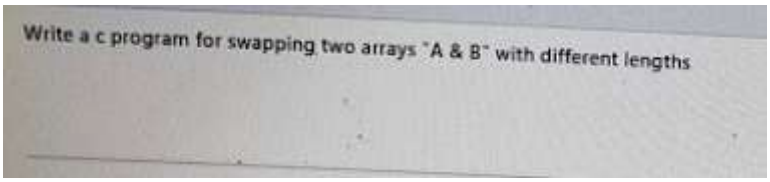


-

التعليقات

```
"include "stdio.h#
()int main
}
;int num, n,new_num

/* Input number from user */
;printf("Enter any number: ")
;fflush(stdin);fflush(stdout)
;scanf("%d", &num)

/* Input bit number you want to clear */
;printf("Enter nth bit to clear (0-31): ")
;fflush(stdin);fflush(stdout)
;scanf("%d", &n)

*/
Left shifts 1 to n times *
Perform complement of above *
finally perform bitwise AND with num and result of above *
/*
;new_num =num & (~(1 << n))

;printf("Bit cleared successfully.\n\n")
;printf("Number before clearing %d bit: %d (in decimal)\n", n, num)
;printf("Number after clearing %d bit: %d (in decimal)\n", n, new_num)

;return 0
{
```

* Q2



-

التعليقات

```
define MAX_SIZE 100 // Maximum array size#
/* Function declarations */
;void inputArray(int arr[], int size)
;void printArray(int arr[], int size)
;void swapArray(int sourceArr[], int destArr[])

()int main
}
;int sourceArr[MAX_SIZE]
;int destArr[MAX_SIZE]
;int size1,size2


Input array size //
;printf("Enter size of array1: ")
;fflush(stdin);fflush(stdout)
;scanf("%d", &size1)

;printf("Enter size of array2: ")
;fflush(stdin);fflush(stdout)
;scanf("%d", &size2)

Input elements of destination array //
;printf("Enter %d elements in source array: ", size1)
;fflush(stdin);fflush(stdout)
;inputArray(sourceArr, size1)

Input element of destination array //
;printf("Enter %d elements in destination array: ", size2)
;fflush(stdin);fflush(stdout)
;inputArray(destArr, size2)


*/
Print elements of both arrays before swapping *
/*
;printf("\n\nSource array before swapping: ")
;printArray(sourceArr, size1)

;printf("\nDestination array before swapping: ")
;printArray(destArr, size2)
```

```
/* .Swap elements of both arrays */
;swapArray(sourceArr, destArr)


*/
Print elements of both arrays after swapping *
/*
;printf("\n\nSource array after swapping: ")
;printArray(sourceArr, size2)

;printf("\nDestination array after swapping: ")
;printArray(destArr, size1)


;return 0
{

void inputArray(int arr[], int size)
}
;int i
for(i=0;i<size;i++)
}
;scanf("%d",&arr[i])
{
{

void printArray(int arr[], int size)
}
;int i
Print elements of array one by one //
for(i=0;i<size;i++)
}
;printf("%d ",arr[i])
{
{
void swapArray(int sourceArr[], int destArr[])
}
;int i
for(i=0;i<MAX_SIZE;i++)
}
write any swapping technique//
;sourceArr[i] ^= destArr[i]
;destArr[i] ^= sourceArr[i]
;sourceArr[i] ^= destArr[i]


{
{
```

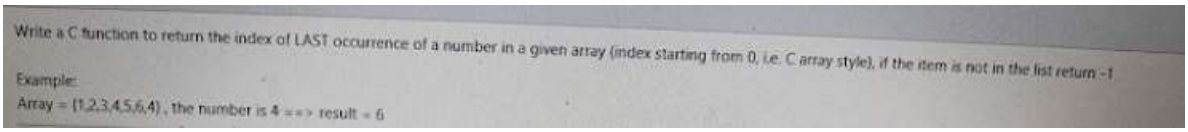
* Q1)Write a C program takes string from the user and check if it the same USERNAME or not.

-

التعليقات

```
int compare(char a[], char b[]);

int main()
{
char a[100]="marwa goda";
char b[100];


printf("please enter your user name\n");
fflush(stdin);fflush(stdout);
gets(b);

if (compare(a, b) == 1)
printf("correct user\n");
else
printf("wrong user\n");

return 0;
}

int compare(char a[], char b[])
{
int i = 0;
while (a[i] == b[i])
{
if (a[i] == '\0' || b[i] == '\0') break;
++i;
}

if (a[i] == '\0' && b[i] == '\0') return 1;
else
return 0;
}
```

* Q8

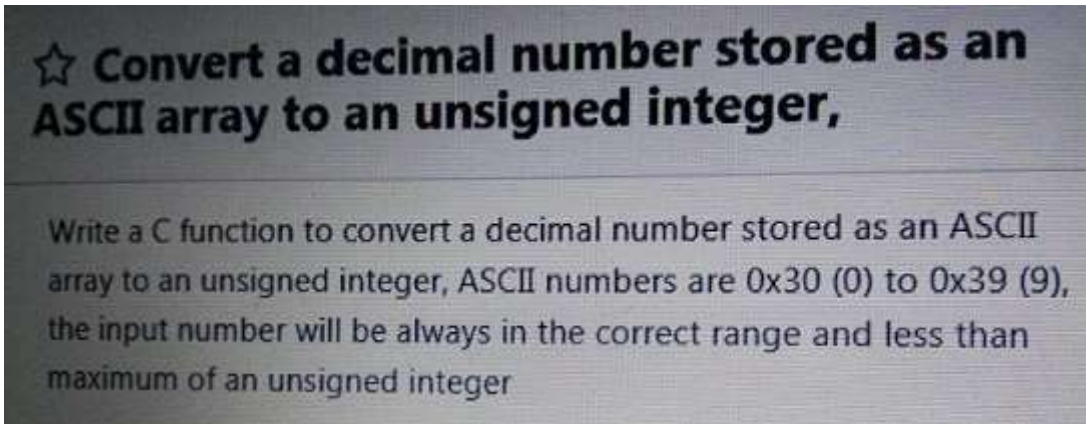


-

التعليقات

```
"include "stdio.h#
;void find_lastindex(int arr[],int num)
()int main
}
;int arr[10]={10,20,5,8,9,20,5,45,90,10}
;int x
;printf("enter your search number ")
;fflush(stdin);fflush(stdout)
;scanf("%d",&x)
;find_lastindex(arr,x)
{
void find_lastindex(int arr[],int num)
}
int i,k=-1;// initialize k=-1 if number not found return -1
for(i=0;i<10;i++)
}
if(num==arr[i])
;k=i
else
;continue

{
;printf("index=%d",k)
{
```

* (Q4



-