Function Quiz

Total points 25/25

Email *

hamed.mohamed26324@gmail.com

0 of 0 points

Name *

Hamed mohamed

Quiz Questions

25 of 25 points

Good luck today! I know you'll do great.

✓ **Q1**) The keyword used to transfer control from a function back to the calling function is

1/1

- switch
- B. goto
- go back
- return

✓ **Q2**) what is the output ? *

1/1

```
#include <stdio.h>
int main()
{
    int i = 5;
    printf("%d %d %d", i++, i++, i++);
    return 0;
}
```

- 7 6 5
- 5 6 7
- 7 7 7
- compiler dependent

✓ **Q3**) Which of the following is true about return type of functions in C? *

- ○ Functions can return any type
- ◉ Functions can return any type except array and functions
- ○ Functions can return any type except array, functions and union
- ○ Functions can return any type except array, functions, function pointer and union

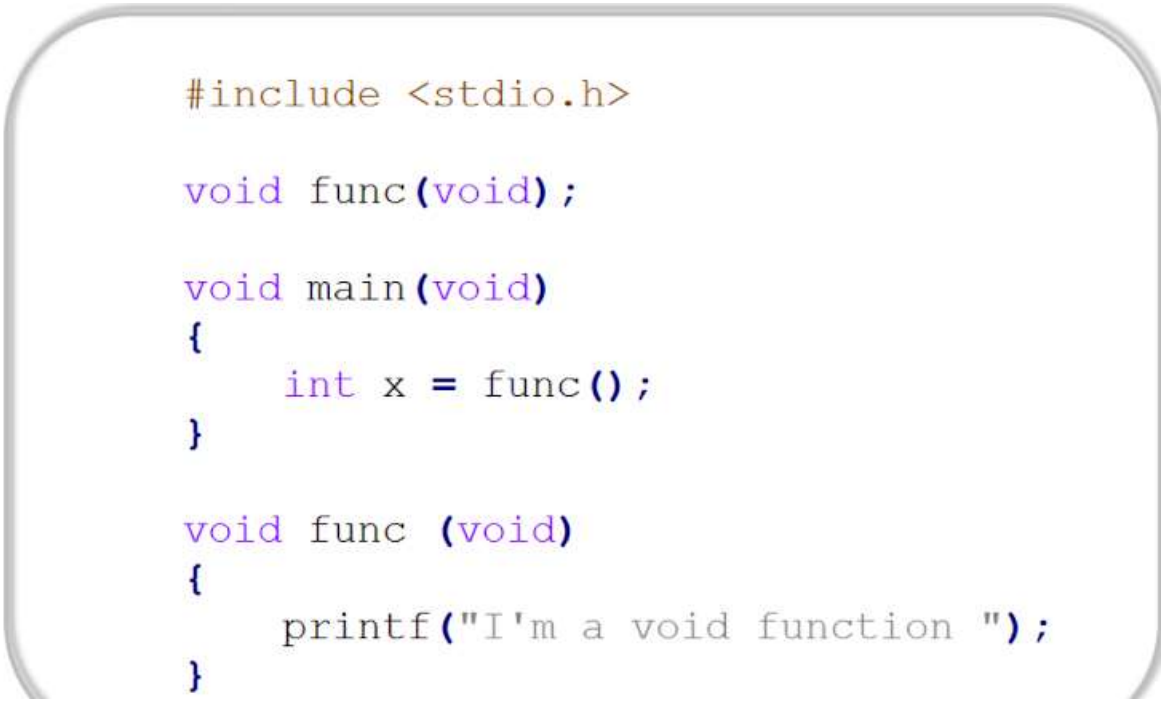
✓ **Q4**) what is the output ? * 2/2

```
int fun()
{
      int i =16 ;
     return i--;
}

int main()
{
  for(fun();fun();fun())
  {
       printf("%d",fun());
  }
     return 0;
}
```

- ○ 13 10 7 4 1
- ○ 15 12 8 5 2
- ◉ infinite loop ✓
- ○ 14 11 8 5 2

✓ **Q5**) what is the output ? * 1/1



```c
#include <stdio.h>

void func(void);

void main(void)
{
    int x = func();
}

void func (void)
{
    printf("I'm a void function ");
}
```

- (●) compilation error ✓
- ( ) I'm a void function
- ( ) No output

✓ **Q6)** If return type for a function is not specified, it defaults to int * 1/1

- (●) True ✓
- ( ) False

✓ **Q7)** what is the output ? * 1/1

```
int fun()
{
    static int i =16 ;
    return i--;
}

int main()
{
 for(fun();fun();fun())
 {
     printf("%d",fun());
 }
    return 0;
}
```

- ○ 13 10 7 4 1
- ○ 15 12 8 5 2
- ○ infinite loop
- ◉ 14 11 8 5 2 ✓

✓ **Q8**) What is the output ? * 1/1

```
#include <stdio.h>

void func(void);

void main(void)
{
    func(10);
}

void func (void)
{
    printf("I'm a void function ");
}
```

- ○ 10
- ○ I'm a void function
- ◉ compilation error ✓

✓ **Q9**)..... is the variable that it is defined inside any function, this variable canbe accessed only on the function that defines it *1/1

- ◉ Local variable ✓
- ○ Global variable
- ○ Static variable
- ○ Extern variable

✓ **Q10)** what is the output ? * 1/1

What will be the output of the program?

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

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

- (●) Hello ✓
- ( ) Hi Hello
- ( ) No output
- ( ) Infinite loop

✓ **Q11)** A function which calls itself is called a ___ function. * 1/1

- ( ) Self Function
- ( ) Auto Function
- (●) Recursive Function ✓
- ( ) Static Function

✓ Q12) * 1/1

19) What is the output of C Program with functions.?

```
void myshow(int);

void main()
{
   int a=10;
   printf("%d ", a);
   myshow(a);
   printf("%d", a);

}

void myshow(int k)
{
   k=20;
}
```

- (●) 10 10 ✓
- ( ) 20 20
- ( ) 10 20
- ( ) Compiler error

✓ **Q13)** * 1/1

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--);
}
```

- ○ Print 5, 4, 3, 2, 1
- ○ Print 1, 2, 3, 4, 5
- ○ Print 5, 4, 3, 2, 1, 0
- ◉ Infinite loop ✓

✓ **Q14)** * 1/1

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;
}
```

- (●) 12, 12 ✓
- ( ) 7, 7
- ( ) 7, 12
- ( ) 12, 7

✓ **Q15)** * 1/1

What will be the output of the program?

```
#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;
}
```

- (●) 5 ✓
- ( ) 4
- ( ) Error
- ( ) Garbage value

✓ **Q16)** * 1/1

What will be the output of the following C code?

```c
#include <stdio.h>
void main()
{
    static int x = 3;
    x++;
    if (x <= 5)
    {
        printf("hi");
        main();
    }
}
```

- ○ Run time error
- ○ hi
- ○ Infinite hi
- ◉ hi hi ✓

✓ **Q17)** * 1/1

What will be the output of the following C code having void return-type function?

```
#include <stdio.h>
void foo()
{
    return 1;
}
void main()
{
    int x = 0;
    x = foo();
    printf("%d", x);
}
```

- ○ 1
- ○ 0
- ○ Runtime error
- ◉ Compile time error ✓

✓ **Q18)** What is the meaning of using extern before function declaration? For example following function sum is made extern *1/1

```
extern int sum(int x, int y, int z)
{
    return (x + y + z);
}
```

- ○ Function is made globally available
- ◉ extern means nothing, sum() is same without extern keyword. ✓
- ○ Function need not to be declared before its use
- ○ Function is made local to the file

✓ **Q19)** What is the meaning of using static before function declaration? *1/1
For example following function sum is made static

```
static int sum(int x, int y, int z)
{
    return (x + y + z);
}
```

- ○ Static means nothing, sum() is same without static keyword.
- ○ Function need not to be declared before its use
- ◉ Access to static functions is restricted to the file where they are declared ✓
- ○ Static functions are made inline

✓ Q20) what is the output ? * 1/1

```c
#include <stdio.h>
void foo(int n, int sum)
{
  int k = 0, j = 0;
  if (n == 0) return;
    k = n % 10;
  j = n / 10;
  sum = sum + k;
  printf ("%d,", k);
  foo (j, sum);

}

int main ()
{
  int a = 2048, sum = 0;
  foo (a, sum);
  printf ("%d\n", sum);

}
```

- ○ 2, 0, 4, 8, 0
- ◉ 8, 4, 0, 2, 0 ✓
- ○ 4, 2, 0 ,8

8, 4, 0, 2, 14

✓ **Q21) Consider the following C code :** *1/1

The below code compiled without any error or warning . If Line 1 is deleted, the below code will show:

```
double foo (double); /* Line 1 */

int main()
{

    double da, db;

    // input da

    db = foo(da);

}

double foo(double a)
{
    return a;
}
```

- ○ no compile warning or error
- ○ some compiler-warnings not leading to unintended results
- ○ some compiler-warnings due to type-mismatch eventually leading to unintended results
- ◉ compiler errors ✓

✓ Q22) what is the output ? * 2/2

```
#include<stdio.h>
int f(int n, int k)
{
    if (n == 0)
        return 0;
    else if (n % 2)
        return f(n/2, 2*k) + k;
    else return f(n/2, 2*k) - k;
}
int main ()
{
    printf("%d", f(20, 1));
    return 0;
}
```

- ○ 5
- ○ 8
- ◉ 9 ✓
- ○ 20

✓ **Q23)** what is the output ? * 1/1

```
#include <stdio.h>
#include <stdbool.h>

void fun(int age);
int main()
{

fun1(22);
 return 0;
}
void fun(int age)
{
    int age ;
    printf("My name is Mohsen and I am %d",age);
}
```

- ○ undefined behaviour
- ○ my name is Mohsen and iam 22
- ○ My name is Mohsen and I am 22
- ◉ compiler error ✓

✓ ***You must know that being here means you are very smart . so be confident in your abilities, and all the best in the final exams ♥.*** *

- ◉ Embedded Systems Circle ✓

Any comments ? *

no thanks.

Quilgo Submission ID (do not edit) *

⚠ DO NOT EDIT this field or your time will not be recorded.

T6FEf34pbVLuAULo