4. Choose the correct answer

1. Which of the following lines of code can be used to reset the least significant bit of x?
   - a. x & 0x01;
   - b. x & ~0x01;
   - c. x|=~0x01;
   - [✓ d. x&= ~0x01;] ✓

2. Type casting is to:
   - a. Convert a lower type to higher type
   - ✓ b. Change the type of the variable
   - c. Obtain the correct value of an Expression
   - ✗ [✓ d. Make an explicit type conversion.]

3. A function prototype is used to
   - a. Define a function
   - [✓ b. Perform compile-time checking when the function is called.] ✓

4. The statement char ch='z' will store in ch
   - a. The character z
   - [✓ b. ASCII value of z]

5. Which of the following is not a C language keyword:
   - ✓ a. sizeof.
   - ✗ [✓ b. case.]
   - c. struct.
   - d. static.

6. Assuming that x is declared as an int, what does the following line do?

   x = x ^ (1<<2)
   - a. Set bit 5 of x
   - b. Clear bit 3 of x
   - ✓ c. Toggle bit 2 of x
   - ✗ [✓ d. Toggle bit 3 of x]

7. How many times "C Programming" is get printed?

```
int main(){
int x;
for(x=-1; x<=10; x++){
    if(x<5)
       continue;
    else
       break;
    printf("C Programming\n");
}
return 0;
}
```

a. Infinite times.

b. 6 times.

c. 12 times.

[✓ d.] 0 times.

8. What will be output when you will execute following c code?

```
void main(){
int check=4;
switch(check){
    default: printf("This is default\n");
    case 1: printf("This is case 1\n");
              break;
    case 2: printf("This is case 2\n");
              break;
    case 3: printf("This is case 3\n");
              break;
}
}
```

[✓ a.] "This is default" then "This is case 1"

b. "This is case 3" then "This is default"

c. "This is default"

d. compilation error.

9. What will be the output of the following code?

```
int Array[5] = {1,2,3,4,5};
void print(int x)
{
   printf("%d \n", x);
}
void main(void)
{
   int i = 3;
   switch(i)
   {
      int y = 9;
      case 1:
         print(1);
         break;
      case 3:
         print(y);
         break;
      default: break;
   }
}
```

9

a. 3

b. 9

c. The code will not compile.

[✓ d.] Garbage.

10. When you run the following piece of code, the output will be:

```
int i = 0;
for(i=10; i>=0; i-=5)
{
   printf("i = %d \t",10-(i-1));
}
```

10 -(10 - 1)

i = 1

i = 6

10 -(5-1)

10 - 4

10 - (0-1)

a. i = 1 i = 5 i = 10

[✓ b.] i = 1 i = 6 i = 11

c. i = 1 i = 6

d. i = 1 i = 2 i = 3

11. Can arrays be passed to the functions by value?

[✓ a. No, Arrays are pointers.]

b. yes

c. No, Arrays can not be passed as arguments.

12. How many times the program will print "C Programming"?

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

a. Infinite times.

b. 32767 times.

c. 65535 times.

[✓ d. Till stack overflows.]

13. The correct syntax to use the typedef for the struct is:

a.
```
typedef struct Mydata
{
   int iData;
}Mydata;
```

b.
```
typedef struct
{
   int iData;
}Mydata;
```

c.
```
struct Mydata
{
   int iData;
};
typedef struct Mydata MyData;
```

[✓ d. All of the Above.]

14.If i = 5, what is the result of the following instructions \

```
if(--i >= 4)
   4 -> 4
printf("Hello");
printf("end");
```

a. hello

b. end

[✓ c.] hello followed by end

15.What will be the output of the following program? 1

```
int main(){
    int i = 4;
    int j = 7;
    j = j || i++ && 1;
    printf("%d %d",i,j);
    return 0;
}
```

$\boxed{i=5}$

j = 7 || 4 && 1

[✓ a.] 5 1

b. 5 7

c. 4 7

~~d.~~ 4 1

16. For a normal C function the compiler ends the function by:

A. transferring the return value to a register.

B. return to the calling function.

What is the correct order of the above two steps?

~~a.~~ A then B.

b. B then A.

[✓ c.] A and B always occur simultaneously.

d. None of the above.

17. A variable P is called pointer to DATA if:

[✓ a.] P contains the address of the first element in DATA.

b. P can store only memory addresses.

c. P contain the DATA and the address of DATA.

18.Assume we defined an array as int arr[20], what will happen in if I write arr[20]

a. Warning.

[✓ b. Compilation error.]

c. Linking error.

19.What is the output of the following code:

```
void main(void){
   unsigned char au8Array[8] = {1,2,3,4,5,6,7,8};
   unsigned int *p;                         00 01 02 03
   p = (unsigned int *)au8Array;                       2
   p++;
   printf("%x\n",*p);
}
```

[✓ a. 0x2]

b. 0x1020304

c. 0x5060708

d. The code will not compile

20.What will be the output of the following code?

```
int main(){
   char arr[10];
   arr = "Hello";
   printf("%s",arr);
   return 0;
}
```

[✓ a. Compiler error.]

b. Linker error.

c. Hello.

21. What will be the output of the code below:

```
int *p;
*p = 20;
```

wild Pointer

a. The code will not compile.

b. The code will run correctly and the value of 20 will be stored without corruption in any memory location.

c. The code will run correctly but the value of 20 will corrupt a memory location.

[✓ d.] The code will cause a run time error.

22. How can you access the element which has the value 4:

```
int arr[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
```

a. arr[0][0]

b. arr[0][1]

[✓ c.] arr[1][0]

d. arr[1][1]

23. What will be output of following program?

```
#include<stdio.h>
int main(){
    unsigned int a = 0x00AADDFF;
    unsigned char *ptr;
    ptr = (unsigned char*) &a;
    printf("%x",*ptr);
    return 0;
}
```

a, b

[✓ a.] FF

b. 00AADDFF

c. 00

d. compiler error

24. Which of the following statements accurately describes the meaning of the declaration int *const x; ?

✓ [✓ a.] x is a constant pointer to integer.

b. x is a pointer to a constant integer.

c. x is a constant integer value.

d. None of the above; it's an invalid C declaration.

25. Which of the following statements accurately describes the intended effect of the declaration int (*a)[10] ?

a. An array of ten integers.

✓ b. A pointer to an array of ten integers.

✗ [✓ c.] An array of ten pointers to integers.

d. An array of ten pointers to function.

26. What will be the output of the following code?

```
int main(){
   unsigned short num1 = 0x00FF;
   unsigned short num2;
   unsigned char *ptr1 = (unsigned char *)&num1;
   unsigned char *ptr2 = (unsigned char *)&num2;
   ptr2[0]=ptr1[1];        F F
   ptr2[1]=ptr1[0];
   printf("%x",num2);
   return 0;
}
```

✗ [✓ a.] 00FF

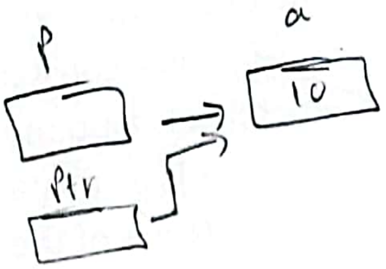
✓ b. FF00

c. Compiler error.

d. Run time error.

27. What will be the output of following code?

```
int main(){
    int a = 10;
    void *p = &a;
    int *ptr = p;
    printf("%d",*ptr);
    return 0;
}
```



a. Address of a

b. Address of p

c. Compiler error

[✓ d.] 10

28. For the code below, select the correct answer.

char *pA, pB;

a. pA and pB are character pointers.

[✓ b.] pA is a character pointer and pB is a character variable.

29. Which of the following is a correct way to write the value 0xAA55 to physical address 0x67A9?

a. uint16_t *p = (uint16_t *)0x67A9; p = 0xAA55;

b. uint16_t *p = (uint16_t *)0xAA55; p = 0x67A9;

[✓ c.] *(uint16_t *const)(0x67A9) = 0xAA55; ✓

d. *(uint16_t *const)(0xAA55) = 0x67A9;

30. The code below implies:

const int *p;

✗ a. Constant pointer.

[✓ b.] Pointer to constant integer.

c. Constant integer.

d. Pointer to integer.

31. What will be the output of this code?

```
#include<stdio.h>
int main(){
   int i = 3;
   int *j;
   int **k;
   j=&i;
   k=&j;
   printf("%u %u %d ",k,*k,**k);
   return 0;
}
```

[✓ a.] Address Address 3

b. Address 3 3

c. 3 3 3

d. Compiler error

32. The main difference between structures and unions is

a. No difference between them.

[✓ b.] Unions conserve memory more that structures.

c. Structures conserve memory more that Unions.

33. In ...... the members share the same memory space

a. Structure.

[✓ b.] Union

c. Array

d. Vector

34. For the code below, select the correct answer:

```
int arr[20];
```

a. (&arr)+1 points to arr[1]

b. (&arr)+1 points to arr[0]

c. (&arr)+1 points to arr[9]

[✓ d.] (&arr)+1 points to arr[20]

35.What will be the output of the following code?

```
 #include <stdio.h>
 int mul(int a, int b, int c){
     return a * b * c;
 }
 int main(){
     int (*ptr)(int, int, int);
     ptr =  mul;
     printf("%d ",ptr(2, 3, 4));
     printf("%d ",(*ptr)(2, 3, 4));
    return 0;
}
```

a. Invalid Operation.

b. Compiler Error.

[✓ c.] 24 24

d. 24 0

36.consider the following definitions in a 32 bit processor(pointers are 32 bits)

```
char * Array[ ] = {"ECP 621","Embedded","Software","Development"};
const char ** ptrArray = &Array;
```

what is the output of: printf("%d",sizeof(Array));

a. 34

b. 44

c. 38

[✓ d.] 16

37.What will be the output of the following code:

```
enum State = {ON, OFF, SLEEP};
enum State aDevice = SLEEP;
aDevice++;
printf("%d",aDevice);
```

[✓ a.] 3

b. ON

c. Compilation error.

d. Undefined behavior.

38.The elements of an array are stored successively in memory cells because:

[✓ a.] By this way computer can keep track only of the address of the first element and the addresses of the other elements can be calculated.

b. The architecture of computer memory does not allow arrays to store other than serially.

39.Passing structure or union by value to a function:

a. Is portable.

b. Results in better performance.

[✓ c.] Causes stack issues.

d. Is supported by all compilers.

40.The C code below will:

```
void main(void)
{
    char *ptr = (char*)malloc(10);
    if(NULL == ptr)
    {
        printf("\n Malloc failed \n");
        return;
    }
    else
    {
        // Do some processing
        free(ptr);
    }
    return;
}
```

[✓ a.] Compile with no warning/errors.

b. Compile with warnings.

c. Not compile.

d. Not pass the preprocessing stage.

Good Luck & Best Wishes ☺
Eng. Mohamed Tarek

Note: the exam consists of 15 **Pages** ☺

1. Write a C function that return number that has the most consecutive occurrence in an array. If more than one number has the same number of consecutive occurrences return the latest.

   Example:

   Array={1,2,2,3,3,3,3,4,4,4,4,3,3}

   Note that 3 and 4 are both consecutive times but is the last occurrence so return 4.

int FindMostOccuring (int Array_Size , int * Array)

2. Write a C function that returns the maximum number of ZEROS between two one's in the number binary.

Example:

Input number = 1010001000001010000001100010100011,

Output = 5

int zeros_max(unsigned int num)

3. Assume that you already have a linked list and head pointer points to the first node in the list. Write a C function that returns the minimum data value inside the list.

```
int getMinimumInsideList(void)
{
```