1. What will the SWAP macro in the following program be expanded to on preprocessing? will the code compile?

```
#include<stdio.h>
#define SWAP(a, b, c)(c t; t=a, a=b, b=t)
int main()
{
    int x=10, y=20;
    SWAP(x, y, int);
    printf("%d %d\n", x, y);
    return 0;
}
```

A. It compiles

B. Compiles with an warning

C. Not compile

D. Compiles and print nothing

Answer: Option **C**
Explanation:
The code won't compile since declaration of `t` cannot occur within parenthesis.
View Answer Discuss in Forum Workspace Report

2. In which stage the following code
`#include<stdio.h>`
gets replaced by the contents of the file `stdio.h`

A. During editing

B. During linking

C. During execution

D. During preprocessing

Answer: Option **D**
Explanation:
The preprocessor replaces the line `#include <stdio.h>` with the system header file of that name. More precisely, the entire text of the file `'stdio.h'` replaces the `#include` directive.

1. What will be the output of the program?

```
#include<stdio.h>
#define MAN(x, y) ((x)>(y)) ? (x):(y);

int main()
{
    int i=10, j=5, k=0;
    k = MAN(++i, j++);
    printf("%d, %d, %d\n", i, j, k);
    return 0;
}
```

A. 12, 6, 12

B. 11, 5, 11

C. 11, 5, Garbage

D. 12, 6, Garbage

Answer: Option **A**
Explanation:
The macro `MAN(x, y) ((x)>(y)) ? (x):(y);` returns the biggest number of given two numbers.
Step 1: `int i=10, j=5, k=0;` The variable `i, j, k` are declared as an integer type and initialized to value 10, 5, 0 respectively.
Step 2: `k = MAN(++i, j++);` becomes,
=> `k = ((++i)>(j++)) ? (++i):(j++);`
=> `k = ((11)>(5)) ? (12):(6);`
=> `k = 12`
Step 3: `printf("%d, %d, %d\n", i, j, k);` It prints the variable `i, j, k`.
In the above macro step 2 the variable `i` value is increemented by 2 and variable `j` value is increemented by 1.

Hence the output of the program is 12, 6, 12

View Answer Discuss in Forum Workspace Report

---

2. What will be the output of the program?

```
#include<stdio.h>
#define SQUARE(x) x*x

int main()
{
    float s=10, u=30, t=2, a;
    a = 2*(s-u*t)/SQUARE(t);
    printf("Result = %f", a);
    return 0;
}
```

A. Result = -100.000000

B. Result = -25.000000

C. Result = 0.000000

D. Result = 100.000000

Answer: Option **A**
Explanation:
The macro function `SQUARE(x) x*x` calculate the square of the given number `'x'`. (Eg: 10^2)
Step 1: `float s=10, u=30, t=2, a;` Here the variable `s, u, t, a` are declared as an floating point type and the variable `s, u, t` are initialized to 10, 30, 2.
Step 2: `a = 2*(s-u*t)/SQUARE(t);` becomes,
=> `a = 2 * (10 - 30 * 2) / t * t;` Here `SQUARE(t)` is replaced by macro to `t*t` .

=> `a = 2 * (10 - 30 * 2) / 2 * 2;`
=> `a = 2 * (10 - 60) / 2 * 2;`
=> `a = 2 * (-50) / 2 * 2 ;`
=> `a = 2 * (-25) * 2 ;`
=> `a = (-50) * 2 ;`
=> `a = -100;`
Step 3: `printf("Result=%f", a);` It prints the value of variable 'a'.

Hence the output of the program is -100

View Answer Discuss in Forum Workspace Report

---

3. What will be the output of the program?

```
#include<stdio.h>
#define SQR(x)(x*x)

int main()
{
    int a, b=3;
    a = SQR(b+2);
    printf("%d\n", a);
    return 0;
}
```

A. 25

B. 11

C. Error

D. Garbage value

Answer: Option **B**
Explanation:
The macro function `SQR(x)(x*x)` calculate the square of the given number `'x'`. (Eg: 10^2)
Step 1: `int a, b=3;` Here the variable `a, b` are declared as an integer type and the variable `b` is initialized to 3.
Step 2: `a = SQR(b+2);` becomes,
=> `a = b+2 * b+2;` Here `SQR(x)` is replaced by macro to `x*x` .
=> `a = 3+2 * 3+2;`
=> `a = 3 + 6 + 2;`
=> `a = 11;`
Step 3: `printf("%d\n", a);` It prints the value of variable 'a'.

Hence the output of the program is 11

View Answer Discuss in Forum Workspace Report

---

4. What will be the output of the program?

```
#include<stdio.h>
#define JOIN(s1, s2) printf("%s=%s %s=%s \n", #s1, s1, #s2, s2);
```

```
int main()
{
    char *str1="India";
    char *str2="BIX";
    JOIN(str1, str2);
    return 0;
}
```

A. str1=IndiaBIX str2=BIX

B. str1=India str2=BIX

C. str1=India str2=IndiaBIX

D. Error: in macro substitution

Answer: Option **B**
Explanation:
No answer description available for this question. Let us discuss.
View Answer Discuss in Forum Workspace Report

5. What will be the output of the program?

```
#include<stdio.h>
#define CUBE(x) (x*x*x)

int main()
{
    int a, b=3;
    a = CUBE(b++);
    printf("%d, %d\n", a, b);
    return 0;
}
```

A. 9, 4

B. 27, 4

C. 27, 6

D. Error

Answer: Option **C**
Explanation:
The macro function `CUBE(x) (x*x*x)` calculates the cubic value of given number(Eg: 10^3.)
Step 1: `int a, b=3;` The variable `a` and `b` are declared as an integer type and varaible `b` id initialized to 3.
Step 2: `a = CUBE(b++);` becomes
=> `a = b++ * b++ * b++;`
=> `a = 3 * 3 * 3;` Here we are using post-increement operator, so the 3 is not incremented in this statement.
=> `a = 27;` Here, 27 is store in the variable `a`. By the way, the value of variable `b` is incremented by 3. (ie: b=6)
Step 3: `printf("%d, %d\n", a, b);` It prints the value of variable `a` and `b`.

Hence the output of the program is 27, 6.

6. What will be the output of the program?

```
#include<stdio.h>
#define PRINT(int) printf("int=%d, ", int);

int main()
{
    int x=2, y=3, z=4;
    PRINT(x);
    PRINT(y);
    PRINT(z);
    return 0;
}
```

A. int=2, int=3, int=4

B. int=2, int=2, int=2

C. int=3, int=3, int=3

D. int=4, int=4, int=4

Answer: Option **A**
Explanation:
The macro `PRINT(int) print("%d,", int);` prints the given variable value in an integer format.
Step 1: `int x=2, y=3, z=4;` The variable x, y, z are declared as an integer type and initialized to 2, 3, 4 respectively.
Step 2: `PRINT(x);` becomes `printf("int=%d,",x)`. Hence it prints 'int=2'.
Step 3: `PRINT(y);` becomes `printf("int=%d,",y)`. Hence it prints 'int=3'.
Step 4: `PRINT(z);` becomes `printf("int=%d,",z)`. Hence it prints 'int=4'.

Hence the output of the program is int=2, int=3, int=4.

View Answer Discuss in Forum Workspace Report

7. What will be the output of the program?

```
#include<stdio.h>
#define SWAP(a, b) int t; t=a, a=b, b=t;
int main()
{
    int a=10, b=12;
    SWAP(a, b);
    printf("a = %d, b = %d\n", a, b);
    return 0;
}
```

A. a = 10, b = 12

B. a = 12, b = 10

C. Error: Declaration not allowed in macro

D. Error: Undefined symbol 't'

Answer: Option **B**

Explanation:

The macro `SWAP(a, b) int t; t=a, a=b, b=t;` swaps the value of the given two variable.
Step 1: `int a=10, b=12;` The variable `a` and `b` are declared as an integer type and initialized to 10, 12 respectively.
Step 2: `SWAP(a, b);`. Here the macro is substituted and it swaps the value to variable `a` and `b`.

Hence the output of the program is 12, 10.

View Answer Discuss in Forum Workspace Report

---

8. What will be the output of the program?

```
#include<stdio.h>
#define FUN(i, j) i##j

int main()
{
    int va1=10;
    int va12=20;
    printf("%d\n", FUN(va1, 2));
    return 0;
}
```

A. 10

B. 20

C. 1020

D. 12

Answer: Option **B**

Explanation:

The following program will make you understand about ## (macro concatenation) operator clearly.

```
#include<stdio.h>
#define FUN(i, j) i##j

int main()
{
    int First  = 10;
    int Second  = 20;

    char FirstSecond[] = "IndiaBIX";

    printf("%s\n", FUN(First, Second) );
    return 0;
}

Output:
-------
IndiaBIX
```

The preprocessor will replace `FUN(First, Second)` as `FirstSecond`.
Therefore, the `printf("%s\n", FUN(First, Second) );` statement will become as `printf("%s\n", FirstSecond );`
Hence it prints `IndiaBIX` as output.
Like the same, the line `printf("%d\n", FUN(va1, 2));` given in the above question will become as `printf("%d\n", va12 );`.
Therefore, it prints `20` as output.
View Answer Discuss in Forum Workspace Report

9. What will be the output of the program?

```
#include<stdio.h>
#define FUN(arg) do\
                 {\
                    if(arg)\
                        printf("IndiaBIX...", "\n");\
                  }while(--i)

int main()
{
    int i=2;
    FUN(i<3);
    return 0;
}
```

A. IndiaBIX...
IndiaBIX...
IndiaBIX

B. IndiaBIX... IndiaBIX...

C. Error: cannot use control instructions in macro

D. No output

Answer: Option **B**

Explanation:

The macro `FUN(arg)` prints the statement "IndiaBIX..." untill the while condition is satisfied.
Step 1: `int i=2;` The variable `i` is declared as an integer type and initialized to 2.
Step 2: `FUN(i<3);` becomes,

```
do
{
    if(2 < 3)
    printf("IndiaBIX...", "\n");
}while(--2)
```

After the 2 while loops the value of i becomes '0'(zero). Hence the while loop breaks.

Hence the output of the program is "IndiaBIX... IndiaBIX..."

View Answer Discuss in Forum Workspace Report

10. What will be the output of the program?

```
#include<stdio.h>
#define MAX(a, b) (a > b ? a : b)

int main()
{
    int x;
    x = MAX(3+2, 2+7);
    printf("%d\n", x);
    return 0;
}
```

A. 8

B. 9

C. 6

D. 5

Answer: Option **B**

Explanation:

The macro `MAX(a, b) (a > b ? a : b)` returns the biggest value of the given two numbers.

Step 1 : `int x;` The variable `x` is declared as an integer type.

Step 2 : `x = MAX(3+2, 2+7);` becomes,

=> `x = (3+2 > 2+7 ? 3+2 : 2+7)`

=> `x = (5 > 9 ? 5 : 9)`

=> `x = 9`

Step 3 : `printf("%d\n", x);` It prints the value of variable `x`.

Hence the output of the program is 9.

11. What will be the output of the program?

```
#include<stdio.h>
#define MIN(x, y) (x<y)? x : y;
int main()
{
    int x=3, y=4, z;
    z = MIN(x+y/2, y-1);
    if(z > 0)
        printf("%d\n", z);
    return 0;
}
```

A. 3

B. 4

C. 0

D. No output

Answer: Option **A**

Explanation:

The macro `MIN(x, y) (x<y)? x : y;` returns the smallest value from the given two numbers.

Step 1: `int x=3, y=4, z;` The variable x, y, z are declared as an integer type and the variable x, y are initialized to value 3, 4 respectively.

Step 2: `z = MIN(x+y/2, y-1);` becomes,

=> `z = (x+y/2 < y-1)? x+y/2 : y - 1;`

=> `z = (3+4/2 < 4-1)? 3+4/2 : 4 - 1;`

=> `z = (3+2 < 4-1)? 3+2 : 4 - 1;`

=> `z = (5 < 3)? 5 : 3;`

The macro return the number 3 and it is stored in the variable `z`.

Step 3: `if(z > 0)` becomes `if(3 > 0)` here the `if` condition is satisfied. It executes the `if` block statements.

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

Hence the output of the program is 3

View Answer Discuss in Forum Workspace Report

12. What will be the output of the program?

```
#include<stdio.h>
#define str(x) #x
#define Xstr(x) str(x)
#define oper multiply

int main()
{
    char *opername = Xstr(oper);
    printf("%s\n", opername);
    return 0;
}
```

A. Error: in macro substitution

B. Error: invalid reference `'x'` in macro

C. print 'multiply'

D. No output

Answer: Option **C**

Explanation:

The macro `#define str(x) #x` replaces the symbol 'str(x)' with 'x'.

The macro `#define Xstr(x) str(x)` replaces the symbol 'Xstr(x)' with 'str(x)'.

The macro `#define oper multiply` replaces the symbol 'oper' with 'multiply'.

Step 1: `char *opername = Xstr(oper);` The varible `*opername` is declared as an pointer to a character type.
=> `Xstr(oper);` becomes,
=> `Xstr(multiply);`
=> `str(multiply)`
=> `char *opername = multiply`
Step 2: `printf("%s\n", opername);` It prints the value of variable `opername`.

Hence the output of the program is "multiply"

View Answer Discuss in Forum Workspace Report

13. What will be the output of the program?

```
#include<stdio.h>
#define MESS junk

int main()
{
    printf("MESS\n");
    return 0;
}
```

A. junk

B. MESS

C. Error

D. Nothing will print

Answer: Option **B**
Explanation:
`printf("MESS\n");` It prints the text "MESS". There is no macro calling inside the printf statement occured.
View Answer Discuss in Forum Workspace Report

14. What will be the output of the program?

```
#include<stdio.h>
#define PRINT(i) printf("%d,",i)

int main()
{
    int x=2, y=3, z=4;
    PRINT(x);
    PRINT(y);
    PRINT(z);
    return 0;
}
```

A. 2, 3, 4,

B. 2, 2, 2,

C. 3, 3, 3,

D. 4, 4, 4,

Answer: Option **A**
Explanation:
The macro `PRINT(i) print("%d,", i);` prints the given variable value in an integer format.
Step 1: `int x=2, y=3, z=4;` The variable x, y, z are declared as an integer type and initialized to 2, 3, 4 respectively.
Step 2: `PRINT(x);` becomes `printf("%d,",x)`. Hence it prints '2'.
Step 3: `PRINT(y);` becomes `printf("%d,",y)`. Hence it prints '3'.
Step 4: `PRINT(z);` becomes `printf("%d,",z)`. Hence it prints '4'.

Hence the output of the program is 2, 3, 4.

View Answer Discuss in Forum Workspace Report

15. What will be the output of the program?

```
#include<stdio.h>
#define MAX(a, b, c) (a>b ? a>c ? a : c: b>c ? b : c)

int main()
{
    int x;
    x = MAX(3+2, 2+7, 3+7);
    printf("%d\n", x);
    return 0;
}
```

A. 5

B. 9

C. 10

D. 3+7

Answer: Option **C**
Explanation:
The macro `MAX(a, b, c) (a>b ? a>c ? a : c: b>c ? b : c)` returns the biggest of given three numbers.
Step 1: `int x;` The variable `x` is declared as an integer type.
Step 2: `x = MAX(3+2, 2+7, 3+7);` becomes,
=> `x = (3+2 >2+7 ? 3+2 > 3+7 ? 3+2 : 3+7: 2+7 > 3+7 ? 2+7 : 3+7)`
=> `x = (5 >9 ? (5 > 10 ? 5 : 10): (9 > 10 ? 9 : 10) )`
=> `x = (5 >9 ? (10): (10) )`
=> `x = 10`
Step 3: `printf("%d\n", x);` It prints the value of 'x'.

Hence the output of the program is "10".

1. Point out the error in the program

```
#include<stdio.h>
#define SI(p, n, r) float si; si=p*n*r/100;
int main()
{
    float p=2500, r=3.5;
    int n=3;
    SI(p, n, r);
    SI(1500, 2, 2.5);
    return 0;
}
```

A. 26250.00 7500.00

B. Nothing will print

C. Error: Multiple declaration of `si`

D. Garbage values

Answer: Option **C**
Explanation:
The macro `#define SI(p, n, r) float si; si=p*n*r/100;` contains the error. To remove this error, we have to modify this macro to
`#define SI(p,n,r) p*n*r/100`
View Answer Discuss in Forum Workspace Report

2. Point out the error in the program

```
#include<stdio.h>

int main()
{
    int i;
    #if A
        printf("Enter any number:");
        scanf("%d", &i);
    #elif B
        printf("The number is odd");
    return 0;
}
```

A. Error: unexpected end of file because there is no matching `#endif`

B. The number is odd

C. Garbage values

D. None of above

Answer: Option **A**
Explanation:
The conditional macro `#if` must have an `#endif`. In this program there is no `#endif` statement written.

1. Which of the following are correct preprocessor directives in C?

   1: #ifdef

   2: #if

   3: #elif

   4: #undef

   A. 1, 2

   B. 4

   C. 1, 2, 4

   D. 1, 2, 3, 4

   Answer: Option **D**

   Explanation:

   The macros `#ifdef #if #elif` are called conditional macros.
   The macro `#undef` undefine the previosly declared macro symbol.

   Hence all the given statements are macro preprocessor directives.

   View Answer Discuss in Forum Workspace Report

---

2. Which of the following are correctly formed `#define` statements in C?

   A. #define CUBE (X) (X*X*X);

   B. #define CUBE(x) (X*X*X)

   C. #define CUBE(X)(X*X*X)

   D. #define CUBE(X) {X*X*X}

   Answer: Option **C**