KillTest

높은 품질, 더 나은 서비스

연습 문제

https://www.killtest.kr

1년간 무료 업데이트 제공

Exam : **CLA-11-03**

Title : CLA - C Certified Associate Programmer

Version : DEMO

1.What happens if you try to compile and run this program?

```
#include <stdio.h>
int *f();
int main (int argc, char *argv[]) {
int *p;
p = f();
printf("%d",*p);
return 0;
}
int *f() {
static v = 1;
return &v;
}
```

Choose the right answer:

A. The program outputs 1

B. Compilation fails

C. The program outputs 3

D. The program outputs 2

E. The program outputs 0

Answer: A

Explanation

The program outputs 1 because the static variable v is initialized to 1 inside the f function, and it is visible to the main function. The f function returns the address of v, which is a pointer to an int. The main function dereferences the pointer and assigns it to p, which is another pointer to an int. Then, the main function prints the value of *p, which is the same as dereferencing p again. Therefore, the output of the program is: f() = &v p = f() printf(“%d”,*p) = &v = 1

The other options are incorrect because they either do not match the output of the program or do not use the correct concept of static variables.

2.What happens if you try to compile and run this program?

```
#define ALPHA 0
#define BETA ALPHA-1
#define GAMMA 1
#define dELTA ALPHA-BETA-GAMMA
#include <stdio.h>
int main(int argc, char *argv[]) {
printf ("%d", DELTA);
return 0;
```

Choose the right answer:

A. The program outputs 2

B. The program outputs -2

C. The program outputs 1

D. Compilation fails

E. The program outputs -1

Answer: D
Explanation
Let's analyze the macros and the program:
1.ALPHA is defined as 0.
2.BETA is defined as ALPHA - 1, which is 0 - 1.
3.GAMMA is defined as 1.
4.DELTA is defined as ALPHA - BETA - GAMMA. With the previous definitions, this expands to 0 - (0 - 1) - 1.
Now, let's expand DELTA with the given values:
makefileCopy code
DELTA=0-(0-1)-1DELTA=0-0+1-1DELTA=0+1-1DELTA=1-1DELTA=0
It is important to note that the macro dELTA is defined with a lowercase 'd', but the printf function is trying to print DELTA with an uppercase 'D'. Preprocessor tokens are case-sensitive, so this is a mismatch. However, for the sake of the question, let's assume that dELTA was meant to be DELTA with an uppercase 'D'.
Since the actual calculation results in 0, but there is a typo in the printf statement (it should print dELTA, not DELTA), the compilation will fail due to DELTA not being defined.

3.What happens if you try to compile and run this program?
```
#include <stdio.h>
int main (int argc, char *argv[]) {
int i =2, j = 1;
if(i / j)
j += j;
else
i += i;
printf("%d",i + j);
return 0;
}
```
Choose the right answer:
A. The program outputs 1
B. The program outputs 5
C. The program outputs 3
D. Compilation fails
E. The program outputs 4
Answer: E
Explanation
In the if statement, i / j is 2 / 1, which is true. Therefore, the if block is executed, and j += j; doubles the value of j (j becomes 2).
After the if-else statement, printf("%d", i + j); prints the sum of i and the updated val-ue of j (2 + 2), which is 4.

4.What happens if you try to compile and run this program?
```
#include <stdio.h>
```

```
#include <string.h>
struct STR {
int i;
char c[20];
float f;
};
int main (int argc, char *argv[]) {
struct STR str = { 1, "Hello", 3 };
printf("%d", str.i + strlen(str.c));
return 0;
}
```

Choose the right answer:
A. The program outputs 4
B. The program outputs 1
C. The program outputs 5
D. The program outputs 6
E. Compilation fails
Answer: D
Explanation
The program defines a structure named STR that contains three members: an int, a char array, and a float. Then it creates a variable of type struct STR named str and initializes its members with the values 1, “Hello”, and 3. The program then prints the value of str.i + strlen(str.c), which is the sum of the int member and the length of the char array member. The length of the string “Hello” is 5, so the expression evaluates to 1 + 5 = 6. Therefore, the program outputs 6. References = C struct (Structures) - Programiz, C Structures (structs) - W3Schools, C Structures - GeeksforGeeks

5.What happens if you try to compile and run this program?

```
#include <stdio.h>
int main (int argc, char *argv[]) {
char *p = "John" " " "Bean";
printf("[%s]", p) ;
return 0;
}
```

Choose the right answer:
A. The program outputs "[]"
B. The program outputs nothing
C. The program outputs [John Bean]
D. The program outputs three lines of text
E. The program outputs two lines of text
Answer: C
Explanation
The string literal "John" " " "Bean" is effectively concatenated into a single string by the compiler during compilation. Therefore, the value of p becomes a pointer to the string "John Bean". The printf statement then prints the string enclosed within square brackets, resulting in the output [John Bean].