

## CS120 Practice Questions

**Disclaimer : Exercises here will help you with your C foundation, but whether or not they end up in exams is another thing. I can't read Edward's 2complex4me mind**

1. Look through the following definitions. If it compiles, Write **C**. Else, write **NC**.

- a. `int foo;` \_\_\_\_\_
- b. `char coo;` \_\_\_\_\_
- c. `short Short;` \_\_\_\_\_
- d. `float int;` \_\_\_\_\_
- e. `float _foo1;` \_\_\_\_\_
- f. `int 2foo;` \_\_\_\_\_
- g. `int do;` \_\_\_\_\_
- h. `int k00;` \_\_\_\_\_
- i. `int b@o;` \_\_\_\_\_
- j. `float define;` \_\_\_\_\_

2. What are the 4 stages which a C file goes through to become an executable?

\_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_

3. Describe what *typedef* does.

\_\_\_\_\_  
\_\_\_\_\_

4. Describe what the pre-processing statement *#define* does.

\_\_\_\_\_  
\_\_\_\_\_

5. What is the difference between using *typedef* and *#define*? Why is it generally better to use *typedef int block* vs *#define int block* ?

---

6. Describe what each of the following gcc compiler flag does:

a. `-ansi`

---

---

b. `-Wall`

---

---

c. `-pedantic`

---

---

d. `-std=c99`

---

---

e. `-O`

---

---

f. `-c`

---

---

7. Read the following code carefully and write the output of the program.

```
#include <stdio.h>

int main(int argc, char** argv)
{
    int i, x = 1;

    for(i = 0; i < 5; ++i)
    {
        switch(++i)
        {
            case 1:
                x++;
            case 2:
                x--;
            case 3:
                x *= 2;
            case 4:
                x /= 2;
            default:
                x = 0;
        }
    }

    printf("x = %d", x);

    return 0;
}
```

Output:

---

8. Read the following code carefully and write the output of the program.

```
#include <stdio.h>

int main(int argc, char** argv)
{
    int i, x = 1;

    for(i = 0; i < 5; ++i)
    {
        switch(i)
        {
            case 1:
                x++;
            case 2:
                x--;
            case 3:
                x *= 2;
            case 4:
                x /= 2;
            default:
                x = 0;
        }
    }

    printf("x = %d", x);

    return 0;
}
```

Output:

---

9. Read the follow code carefully.

```
int i, sum=0;

int main(int argc, char** argv)
{
    if(argc < 2)
    {
        printf("2 arguments or more!\n");
        return 0;
    }
    for(i=0; i < argc; ++i)
    {
        printf(argv[i]);
    }
    return 95;
}
```

Does the code compile?

If not, explain why and fix it. If it does, explain what the code does.

---

---

---

---

10. Read the following code carefully and write the output of the program.

```
#include <stdio.h>

int main(int argc, char** argv)
{
    int i = 0, j = 5, k = 1, l = 2;
    int x;

    if(i==1)
    {
        x = i;
    }
    else if(--k || --j-++l == k)
    {
        x = 99;
    }
    printf("i = %d, j = %d, k = %d, l = %d, x = %d", i, j, k, l, x);

    return 0;
}
```

Output:

---

11. Read the following code carefully and write the output of the program.

```
#include <stdio.h>

int main(int argc, char** argv)
{
    int i = 0, j = 5, k = 1, l = 2;
    int x;

    if(i++==1)
    {
        x = i;
    }
    else if(k && --j-++l == k || k++ && i++)
    {
        x = 99;
    }
    printf("i = %d, j = %d, k = %d, l = %d, x = %d", i, j, k, l, x);

    return 0;
}
```

Output:

---