

1. What will be the output of the following C code?

```
#include <stdio.h>

void gello()
{
    printf("Gello ");
}

void hello()
{
    gello();
    printf("Hello ");
}

int main()
{
    printf("Main ");
    hello();
}
```

- Main Hello Gello
- Main Gello
- Main Gello Hello
- Main Hello

Explanation: প্রথমে main() ফাংশন থেকে "Main" প্রিন্ট হয়। তারপর hello() কল হয়, যা gello() ফাংশন কে কল করে, gello() ফাংশন থেকে "Gello" প্রিন্ট করে। পরে gello() ফাংশন থেকে রিটোর্ন করে hello() ফাংশন থেকে "Hello" প্রিন্ট হয়। সুতরাং আউটপুট হবে Main Gello Hello

2. “A function is calling itself” What does this process called?

- Call by value.
- Call by name.
- Recursion.
- Call by reference.

Explanation: যখন কোনো ফাংশন নিজেকে নিজেই কল করে, তখন সেই প্রক্রিয়াকে রিকার্শন (Recursion) বলে। এটি একটি লুপের মতো কাজ করে এবং একটি বেস কেসের মাধ্যমে থামে।

3. What will be the problem if we run this code?

```
#include <stdio.h>
void fun()
{
    printf("fun\n");
    fun();
}
int main()
{
    fun();
}
```

- Syntax error.
- Compilation error.
- Infinite Recursion.
- Both a and b.

Explanation: এই কোডে `fun()` ফাংশন বারবার নিজেকে কল করে কোনো Base Case ছাড়াই, যার ফলে এটি ইনফিনিট রিকার্শনে (Infinite Recursion) চলতে থাকে।

4. What will be the output, if we run this code?

```
#include <stdio.h>
void fun()
{
    printf("fun\n");
}
int main()
{
    fun();
}
```

- Syntax error.
- Compilation error.
- fun
- No Output

Explanation: এই কোডে main() ফাংশন fun() কে কল করে, যা "fun" প্রিন্ট করে। কোনো সিনট্যাক্স বা কম্পাইলেশন এর নেই, তাই আউটপুট হবে শুধু fun

5. What will be the output of the following C code?

```
#include <stdio.h>

void f2()
{
    printf("f1 ");
}

void f1()
{
    printf("f2 ");
    f2();
}

int main()
{
    printf("Main ");
    f1();
}
```

- Main f1 f2
- Main f2 f1
- Main
- No output.

Explanation: প্রথমে main() থেকে "Main" প্রিন্ট হয়। তারপর f1() কল হয়, যা "f2" প্রিন্ট করে এবং f1() ফাংশনের ভিতরে f2() কল করে যা "f1" প্রিন্ট করে। সুতরাং আউটপুট হবে: Main f2 f1

6. Recursion is similar but not exact to which of the following?

- If-else
- loop
- Both of the above
- None of the above

Explanation: রিকার্শন এবং লুপ উভয়ই পুনরাবৃত্তিমূলক কাজ করে, তবে রিকার্শন ফাংশন কল ব্যবহার করে, আর লুপ কন্ডিশন চেক করে। এদের কাজের ধারণা একই রকম, কিন্তু ইমপ্লিমেন্টেশন ভিন্ন। তাই রিকার্শন লুপের মতো, কিন্তু হ্রবহ এক নয়।

7. Can you print from N to 1 and 1 to N using recursion?

- Yes
- No

Explanation: রিকার্শন ব্যবহার করে N থেকে 1 এবং 1 থেকে N প্রিন্ট করা সম্ভব। (যা মডিউলে ভালোভাবে বুঝানো হয়েছে)

8. A recursive function must always have a base case to prevent infinite recursion.

- False
- True

Explanation: বেস কেস ছাড়া ফাংশন নিজেকে infinite times কল করতে থাকবে, ফলে স্ট্যাক ওভারফ্লো (Stack Overflow) এর হবে এবং প্রোগ্রাম ক্র্যাশ করবে। তাই, একটি রিকার্শন ফাংশনে অবশ্যই একটি বেস কেস (Base Case) থাকা প্রয়োজন, যাতে রিকার্শন stop হয়।

9. In recursion, the condition for which the function will stop calling itself is?

- Best case
- Worse case
- Base case
- Nothing

Explanation: রিকার্শনে ফাংশন নিজেকে কল করা বন্ধ করার শর্তকে বেস কেস (Base Case) বলে।

10. A base case is not necessary for a recursive function to work correctly.

- True
- False

Explanation: বেস কেস (Base Case) ছাড়া রিকার্শন ফাংশন সঠিকভাবে কাজ করতে পারে না। কারণ বেস কেসই রিকার্শনকে থামানোর শর্ত দেয়। এটি না থাকলে ফাংশন নিজেকে অসীমভাবে (infinitely) কল করতে থাকবে। সুতরাং, এই কুইজের উত্তর False হবে।