

ট্রিকি কুইজগুলো এখানে এক্সপ্লেন করে দেওয়া আছে। তারপরও কোথাও ডাউট থাকলে কুইজের উত্তর শো করে পোস্ট না করে সাপোর্ট সেশনে জয়েন হয়ে ক্লিয়ার হয়ে নিয়োন।

1. What is a pointer in C programming?

- a) A variable that can only store integer values
- b) A variable that stores the value of another variable
- c) A variable that stores the memory address of another variable**
- d) A variable that can only store character values

Explanation: পয়েন্টার হচ্ছে একটি ভেরিয়েবল যা অন্য একটি ভেরিয়েবল এর এড্রেস স্টোর করে।

2. Which operator is used to access the value at the address pointed to by a pointer in C programming?

- a) ***
- b) %
- c) ^
- d) &

Explanation: * সাইন দিয়ে আমরা পয়েন্টার যেই ভেরিয়েবলকে পয়েন্ট করে আছে তার ভেলু এক্সেস করি।

3. What will be the output of the following code snippet?

```
int x = 10;
int *p = &x;
x++;
printf("%d", *p);
```

- a) 10
- b) 11**
- c) 9
- d) Garbage value

Explanation:

`int x = 10;` // x এর ভেলু ১০

`int *p = &x;` // p পয়েন্টার x কে পয়েন্ট করছে

`x++;` // x এর ভেলু ১ বাড়িয়ে এখন ১১ হলো।

`printf("%d", *p);` // p পয়েন্টার যেই ভেরিয়েবল কে পয়েন্ট করছে তার ভেলু প্রিন্ট করা হচ্ছে। অর্থাৎ x এর ভেলু প্রিন্ট করা হচ্ছে। যার ভেলু ১১। তাই ১১ প্রিন্ট হবে।

4. What will be the output of the following code snippet?

```
int x = 10;
int *p = &x;
int y = 20;
*p = ++y;
```

```
x++;  
printf("%d",x);
```

- a) 11
- b) 20
- c) 21
- d) 22

Explanation:

```
int x = 10;    // x এর ভেলু ১০  
int *p = &x;  // p পয়েন্টার x কে পয়েন্ট করছে  
int y = 20;    // y এর ভেলু ২০  
*p = ++y;      // প্রি ইনক্রিমেন্ট করে আগে y এর মান ১ বাড়িয়ে ২১ করা হল। তারপর পয়েন্টার p এর ভেলু চেঞ্জ  
করে y এর মান অর্থাৎ ২১ করে দেওয়া হচ্ছে। যেহেতু p পয়েন্টার x কে পয়েন্ট করছিল তাই এখন x এর ভেলুও হচ্ছে ২১  
x++;           // x এর ভেলু ১ বাড়ানো হচ্ছে। তাই এখন x এর ভেলু ২২।  
printf("%d",x);    // ২২ প্রিন্ট হবে।
```

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

```
int x = 100;  
int *p = &x;  
printf("%d\n",p);
```

- a) 100
- b) **Address of variable x.**
- c) Address of pointer x.
- d) Garbage value.

Explanation: p একটি পয়েন্টার যা x কে পয়েন্ট করছে অর্থাৎ x এর এড্রেসকে স্টোর করছে। তাই p প্রিন্ট করলে x এর এড্রেস প্রিন্ট হবে।

6. Look at the following code snippet.

```
int a[4] = {1, 2, 3, 4};  
int *p;  
p = a + 2;  
p++;  
*p = 100;
```

Now, if we print the array, what will be the output?

- a) 1 2 3 4
- b) 1 100 3 4
- c) 1 2 100 4
- d) **1 2 3 100**

Explanation:

```
int a[4] = {1, 2, 3, 4};
```

```
int *p;
```

```
p = a + 2; // পয়েন্টারে এরের ইন্ডেক্স ২ এর ভেলুকে রাখা হচ্ছে। অর্থাৎ পয়েন্টার এখন ভেলু ৩ কে পয়েন্ট করছে।
```

```
p++; // পয়েন্টার কে ১ বাড়ানো হচ্ছে। অর্থাৎ পয়েন্টার এখন পরবর্তী ভেলু ৪ কে পয়েন্ট করছে
```

```
*p = 100; // পয়েন্টার যাকে পয়েন্ট করছে তার ভেলু চেঞ্জ করে ১০০ দিয়ে দেওয়া হচ্ছে। অর্থাৎ ভেলু ৪ চেঞ্জ হয়ে ১০০ হয়ে গেল। তাই এখন এরে হচ্ছে ১,২,৩,১০০
```

7. Look at the following code snippet?

```
int x = 10;
```

```
int *p = &x;
```

```
printf("%d",p);
```

After running this code we got an output. Now how can we get the same output?

a) `printf("%d",x);`

b) `printf("%d",*p);`

c) `printf("%d",&p);`

d) `printf("%d",&x);`

Explanation: উপরের কোডটিতে আমরা পয়েন্টার কে প্রিন্ট করছি। পয়েন্টার এ আমরা x ভেরিয়েবল এর এড্রেসকে প্রিন্ট করেছিলাম। তাই পয়েন্টার প্রিন্ট করলে x এর এড্রেস প্রিন্ট হবে।

তাই সঠিক উত্তর হচ্ছে লাস্ট অপশন। লাস্ট অপশন প্রিন্ট করলে x এর এড্রেস প্রিন্ট হবে।

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

```
int a[5] = {10, 20, 30, 40, 50};
```

```
printf("%d", *a);
```

a) Error

b) Address of index number 0.

c) 10

d) Garbage value.

Explanation: এরে আসলে মেমরিতে একটি পয়েন্টার হিসেবে কাজ করে। যার মধ্যে এরের প্রথম ভেলু অর্থাৎ ইন্ডেক্স ০ এর এড্রেস স্টোর থাকে। তাই সেই পয়েন্টার এর ভেলু প্রিন্ট করলে ইন্ডেক্স ০ এর ভেলু ১০ প্রিন্ট হবে।

9. Which function did we use today to declare a dynamic array?

a) new array

b) malloc()

c) `data_type array_name[size]`

d) None of the above

Explanation: malloc() ফাংশন ইউজ করেছি আজকে আমরা।

10. What will be the output of the following code snippet? [Tricky question. Don't ask without thinking]

```
int x,y,z;  
x = 10, y = 20, z = 30;  
int *p, *q;  
p = &x;  
q = &y;  
*p = 40;  
x = 50;  
q = p;  
printf("%d",*q);
```

- a) 20
- b) 30
- c) 40
- d) 50

Explanation:

```
int x,y,z;  
x = 10, y = 20, z = 30;  
int *p, *q;  
p = &x; // একটি পয়েন্টার p নিয়ে x এর এড্রেস স্টোর করছি।  
q = &y; // একটি পয়েন্টার q নিয়ে y এর এড্রেস স্টোর করছি।  
*p = 40; // p পয়েন্টার যেই ভেরিয়েবল কে পয়েন্ট করছে তার ভেলু চেঞ্জ করে ৪০ করে দেওয়া হলো।  
x = 50; // x এর ভেলু ৫০ করে দেওয়া হলো। এক্ষেত্রে p কিন্তু x কে পয়েন্ট করছে।  
q = p; // পয়েন্টার q কে p এর সমান করে দেওয়া হচ্ছে। অর্থাৎ পয়েন্টার qও এখন x ভেরিয়েবলকে পয়েন্ট করবে।  
printf("%d",*q); // q পয়েন্টার x ভেরিয়েবলকে পয়েন্ট করছে তাই x এর ভেলু অর্থাৎ ৫০ প্রিন্ট হবে।
```

কুইজে ভুল হলে হতাশ হওয়ার কিছু নেই। ভুল করেছেন মানেই নতুন কিছু শিখেছেন। আপনি যদি কুইজ দিয়ে ১০/১০ পান। তারমানে আপনার নতুন কিছু শিখা হয় নি। মডিউল দেখে যা শিখেছেন সেটাই জাস্ট যাচাই করলেন এখানে।

কিন্তু আপনি যদি ৮/১০ পান অথবা ৭/১০ পান তারমানে আপনি ২টা বা ৩টা ভুল করেছেন। তারপর আপনি যখন চেক করে দেখছেন যে কেন ভুল হলো এবং সেই ভুল থেকে শিক্ষা নিচ্ছেন। তখন আপনার নতুন জিনিস শিখা হলো যেটা আপনার মডিউল দেখে হয়নি।

তাই ভুল করতে থাকুন। শিখতে থাকুন। হ্যাপি কোডিং 😊

কুইজ এবং এক্সপ্লেনেশন ডক আপনাদের কেমন লেগেছে তা আমাদের জানাতে পারেন এখানে: [feedback form](#)