

۳۴- دستور توقف دستورالعمل‌های تکرار کدام است؟

الف- end

ب- stop

ج- break

د- {

توضیح: دستور break در C++ برای توقف دستورالعمل‌های تکرار (حلقه‌ها) استفاده می‌شود. این دستور باعث خروج فوری از حلقه می‌شود. به عنوان مثال:

```
#include <iostream>
```

```
int main() {  
    for (int i = 0; i < 10; ++i) {  
        if (i == 5) {  
            break; // خروج از حلقه زمانی که  
        }  
        std::cout << i << " ";  
    }  
    return 0;  
}
```

در این مثال، زمانی که مقدار i به ۵ می‌رسد، دستور break اجرا می‌شود و حلقه for متوقف می‌شود. نتیجه‌ی این برنامه چاپ اعداد ۰ تا ۴ خواهد بود.

۳۵- کدام گزینه تعریف درست یک آرایه می‌باشد؟

الف- int x[5]

ب- 5(int

ج- 5(int x

د- x[5]String

توضیح: برای تعریف یک آرایه در C++، باید نوع داده، نام آرایه و اندازه آرایه را مشخص کنید. فرم صحیح تعریف یک آرایه به صورت زیر است:

با ۵ عنصر `int` یک آرایه از نوع `int x[5];` //

در این مثال، `int x[5]` یک آرایه به نام `x` از نوع `int` با ۵ عنصر تعریف می‌کند. هر یک از عناصر این آرایه می‌تواند یک مقدار صحیح (`integer`) ذخیره کند. سایر گزینه‌ها فرم صحیح تعریف آرایه را ندارند.

۳۶- آرایه دو بعدی روبه‌رو چند سطر و ستون دارد؟ `int a[3][4];`

الف- چهار سطر و سه ستون

ب- سه سطر و چهار ستون

ج- سه سطر و سه ستون

د- چهار سطر و چهار ستون

توضیح: تعریف آرایه دو بعدی `int a[3][4];` نشان می‌دهد که این آرایه دارای ۳ سطر و ۴ ستون است. اولین عدد (۳) تعداد سطرها و دومین عدد (۴) تعداد ستون‌ها را مشخص می‌کند. به عنوان مثال، می‌توان این آرایه را به صورت زیر تجسم کرد:

`a[0][0]`, `a[0][1]`, `a[0][2]`, `a[0][3]`
`a[1][0]`, `a[1][1]`, `a[1][2]`, `a[1][3]`
`a[2][0]`, `a[2][1]`, `a[2][2]`, `a[2][3]`

بنابراین، این آرایه دارای ۳ سطر و ۴ ستون است.

۳۷- شروع شماره سطر و ستون از چه مقدار است؟

الف- ۰

ب- ۱

ج- سطر ۰ و ستون ۱

د- سطر ۱ و ستون ۰

توضیح: در C++، شماره‌گذاری سطر و ستون در آرایه‌ها از مقدار ۰ شروع می‌شود. این بدان معناست که اولین عنصر در یک آرایه یک‌بعدی `a` با `a[0]` مشخص می‌شود. به همین ترتیب، در یک آرایه دوبعدی مانند `int a[3][4]`، شماره‌گذاری سطرها و ستون‌ها به صورت زیر است:

سطرها: ۰، ۱، ۲

ستونها: ۰، ۱، ۲، ۳

به عنوان مثال:

```
int a[3][4];  
a[0][0] = 1; // اولین سطر و اولین ستون  
a[2][3] = 10; // سومین سطر و چهارمین ستون
```

بنابراین، شماره گذاری سطر و ستون در آرایه ها از ۰ شروع می شود.

۳۸- در توابع برگشتی کدام حالت نیاز به نوشتن پیش نمونه تابع نمی باشد؟

الف- int

ب- void

ج- char

د- null

توضیح: در زبان C++، اصطلاح "null" به عنوان نوع بازگشتی تابع معتبر نیست. با این حال، هیچ یک از گزینه های دیگر نیز به خودی خود مستلزم حذف پیش نمونه تابع (function prototype) نیستند.

در C++، هر تابعی که در بدنه ی تابع main یا هر تابع دیگری استفاده می شود، باید پیش نمونه اش (prototype) قبل از استفاده تعریف شود مگر اینکه تابع قبل از استفاده تعریف شده باشد. این شامل توابع با نوع بازگشتی int، void و char می شود. به عنوان مثال:

```
#include <iostream>  
  
// پیش نمونه تابع  
void myFunction();  
  
int main() {  
    myFunction();  
    return 0;  
}  
  
// تعریف تابع
```

```
void myFunction() {
    std::cout << "Hello, World!" << std::endl;
}
```

با این حال، اگر تابع قبل از استفاده آن تعریف شود، نیازی به پیش‌نمونه نیست:

```
#include <iostream>
```

```
void myFunction() {
    std::cout << "Hello, World!" << std::endl;
}
```

```
int main() {
    myFunction();
    return 0;
}
```

در این مثال، تابع `myFunction` قبل از فراخوانی آن در `main` تعریف شده است، بنابراین نیازی به پیش‌نمونه تابع نیست. به عبارت دیگر، نوع بازگشتی تابع تعیین نمی‌کند که آیا پیش‌نمونه نیاز است یا خیر؛ بلکه ترتیب تعریف توابع و استفاده از آن‌ها این نیاز را تعیین می‌کند.

۳۹- رشته‌ها در `C++` به کدام عبارت ختم می‌شوند؟

الف- `0/`

ب- `\`

ج- `\0`

د- `{`

توضیح: در `C++`، رشته‌های کاراکتری (C-style strings) به یک کاراکتر `null` (یعنی `\0`) ختم می‌شوند. این کاراکتر `null` نشان‌دهنده پایان رشته است و به تابع‌های کتابخانه‌ای اجازه می‌دهد که تشخیص دهند رشته تا کجا ادامه دارد.

به عنوان مثال:

```
char str[] = "Hello";
```

در این کد، آرایه str شامل شش عنصر است: H, e, l, l, o و \0. کاراکتر null در انتهای رشته قرار می‌گیرد تا نشان دهد که رشته به پایان رسیده است.

۴۰- کدام تابع برای مقایسه دو رشته استفاده می‌شود؟

الف- strcmp(s1, s2)

ب- compare(s1, s2)

ج- different(s1, s2)

د- stringcmp(s1, s2)

توضیح: برای مقایسه دو رشته در C++ از تابع strcmp استفاده می‌شود. این تابع در فایل سرآیند <cstring> تعریف شده است و دو رشته را با هم مقایسه می‌کند. اگر رشته‌ها برابر باشند، مقدار ۰ برمی‌گرداند؛ اگر رشته اول از نظر لغوی کمتر از رشته دوم باشد، مقدار منفی و اگر بیشتر باشد، مقدار مثبت برمی‌گرداند.

به عنوان مثال:

```
#include <iostream>
```

```
#include <cstring>
```

```
int main() {
```

```
    const char* str1 = "Hello";
```

```
    const char* str2 = "World";
```

```
    int result = strcmp(str1, str2);
```

```
    if (result == 0) {
```

```
        std::cout << "The strings are equal." << std::endl;
```

```
    } else if (result < 0) {
```

```
        std::cout << "The first string is less than the  
second string." << std::endl;
```

```
    } else {
```