

集训课-23: C++高中组

课程目标

01

单选题

02

编程题



Part 1

单选题

单选题

1、下列对初始化字符串的方式描述不正确的是（ ）。

A 默认初始化： `string s;`

B 直接初始化： `string s1="hello world";`

C 拷贝初始化： `string s2=s1;`

D 使用单个字符初始化： `string s(5, 'a');`

单选题

1、下列对初始化字符串的方式描述不正确的是（ ）。

A 默认初始化: `string s;`

B 直接初始化: `string s1="hello world";`

C 拷贝初始化: `string s2=s1;`

D 使用单个字符初始化: `string s(5, 'a');`

单选题

2、执行下列程序，输出的结果是（ ）。

```
#include <iostream>
using namespace std;
int main()
{
    string str = "size";
    cout << str.length() << endl;
    return 0;
}
```

A 4

B 3

C 2

D 1

单选题

2、执行下列程序，输出的结果是（ ）。

```
#include <iostream>
using namespace std;
int main()
{
    string str = "size";
    cout << str.length() << endl;
    return 0;
}
```

A 4

B 3

C 2

D 1

单选题

3、下列对字符数组进行初始化的语句中，正确的是（ ）。

A char a[] ="Hello";

B char a[5]={'Hello'};

C char a[5]= "Hello";

D char a[5]={"Hello"};

单选题

3、下列对字符数组进行初始化的语句中，正确的是（ ）。

A `char a[] = "Hello";`

B `char a[5]={'Hello'};`

C `char a[5]= "Hello";`

D `char a[5]={ "Hello"};`

单选题

4、下列对字符数组进行初始化的语句中，正确的是（ ）。

A char array4[];

B char array2[10] = {0};

C char array4[][10];

D char array3[2] = "hello";

单选题

4、下列对字符数组进行初始化的语句中，正确的是（ ）。

A char array4[];

B char array2[10] = {0};

C char array4[][10];

D char array3[2] = "hello";

单选题

5、下面对于穷举算法描述不正确的是()。

- A 穷举算法是蛮力策略的具体体现
- B 穷举算法常用于解决“是否存在”或“有多少种可能”等问题
- C 穷举算法在实际应用中，通常用来解决规模较大的问题
- D 穷举算法的特点是算法设计比较简单

单选题

5、下面对于穷举算法描述不正确的是()。

A 穷举算法是蛮力策略的具体体现

B 穷举算法常用于解决“是否存在”或“有多少种可能”等问题

C 穷举算法在实际应用中，通常用来解决规模较大的问题

D 穷举算法的特点是算法设计比较简单

单选题

6、下列函数的功能是进行排序，其使用的排序算法是（ ）。

```
void Sort(int* h,int* len){  
    if(h==NULL) return;  
    if(len<=1) return;  
    for(int i=0;i<len-1;++i)  
        for(int j=0;j<len-1-i;++j)  
            if(h[j]>h[j+1]){  
                int t = h[j];  
                h[j] = h[j+1];  
                h[j+1] = t;  
            }  
    return;  
}
```

A 选择排序

B 堆排序

C 快速排序

D 冒泡排序

单选题

6、下列函数的功能是进行排序，其使用的排序算法是（ ）。

```
void Sort(int* h,int* len){  
    if(h==NULL) return;  
    if(len<=1) return;  
    for(int i=0;i<len-1;++i)  
        for(int j=0;j<len-1-i;++j)  
            if(h[j]>h[j+1]){  
                int t = h[j];  
                h[j] = h[j+1];  
                h[j+1] = t;  
            }  
    return;  
}
```

A 选择排序

B 堆排序

C 快速排序

D 冒泡排序

单选题

7、执行下列代码，当输入4时，输出的结果是（ ）。

```
#include <iostream>
using namespace std;
long factorial(int value);
int main() {
    int value;
    cout << "input number: ";
    cin >> value;
    cout << factorial(value) << endl;
    return 0;
}
long factorial(int value) {
    if (value == 0) {
        return 1;
    }
    return value + factorial(value - 1);
}
```

A 10

B 11

C 24

D 26

单选题

7、执行下列代码，当输入4时，输出的结果是（ ）。

```
#include <iostream>
using namespace std;
long factorial(int value);
int main() {
    int value;
    cout << "input number: ";
    cin >> value;
    cout << factorial(value) << endl;
    return 0;
}
long factorial(int value) {
    if (value == 0) {
        return 1;
    }
    return value + factorial(value - 1);
}
```

A 10

B 11

C 24

D 26

单选题

8、下列函数完成的功能是（ ）。

```
int BinarySearch(int* pArray, int nBegin, int nEnd, int nKey){  
    if (NULL == pArray || nBegin > nEnd){  
        return -1;  
    }  
    int nMid = (nBegin + nEnd) / 2;  
    if (pArray[nMid] == nKey){  
        return nMid;  
    }  
    else if (pArray[nMid] > nKey){  
        return BinarySearch(pArray, nBegin, nMid - 1, nKey);  
    }  
    else{  
        return BinarySearch(pArray, nMid + 1, nEnd, nKey);  
    }  
}
```

- A 通过递归实现贪心算法
- B 通过非递归实现贪心算法
- C 通过非递归实现二分算法
- D 通过递归实现二分算法

单选题

8、下列函数完成的功能是（ ）。

```
int BinarySearch(int* pArray, int nBegin, int nEnd, int nKey){  
    if (NULL == pArray || nBegin > nEnd){  
        return -1;  
    }  
    int nMid = (nBegin + nEnd) / 2;  
    if (pArray[nMid] == nKey){  
        return nMid;  
    }  
    else if (pArray[nMid] > nKey){  
        return BinarySearch(pArray, nBegin, nMid - 1, nKey);  
    }  
    else{  
        return BinarySearch(pArray, nMid + 1, nEnd, nKey);  
    }  
}
```

- A 通过递归实现贪心算法
- B 通过非递归实现贪心算法
- C 通过非递归实现二分算法
- D 通过递归实现二分算法**

单选题

9、下列函数通过非递归的方式实现二分算法，横线处应补上的内容是（ ）。

```
int BinarySearch(int array[], int len, int value)
{
    if (array == NULL || len <= 0)
        return -1;
    int low = 0;
    int high = len - 1;
    while (low <= high)
    {
        int mid = low + (high - low) / 2;
        if (array[mid] == value)
            return mid;
        else if ( _____ )
            high = mid - 1;
        else
            low = mid + 1;
    }
    return -1;
}
```

A array[low] > value

B array[high] < value

C array[mid] > value

D array[mid] < value

单选题

9、下列函数通过非递归的方式实现二分算法，横线处应补上的内容是（ ）。

```
int BinarySearch(int array[], int len, int value)
{
    if (array == NULL || len <= 0)
        return -1;
    int low = 0;
    int high = len - 1;
    while (low <= high)
    {
        int mid = low + (high - low) / 2;
        if (array[mid] == value)
            return mid;
        else if ( _____ )
            high = mid - 1;
        else
            low = mid + 1;
    }
    return -1;
}
```

A array[low] > value

B array[high] < value

C array[mid] > value

D array[mid] < value

单选题

10、下列函数通过选择排序的方式将数组元素从小到大排序，横线处应补上的语句是（ ）。

```
void sort(int* a, int n) {  
    for (int i = 1; i < n; ++i) {  
        int ith = i;  
        for (int j = i + 1; j <= n; ++j) {  
            if ( _____ ) {  
                ith = j;  
            }  
        }  
        int t = a[i];  
        a[i] = a[ith];  
        a[ith] = t;  
    }  
}
```

A $a[j] > a[j+1]$

B $a[j+1] > a[ith]$

C $a[j] < a[ith]$

D $a[j] > a[ith]$

单选题

10、下列函数通过选择排序的方式将数组元素从小到大排序，横线处应补上的语句是（ ）。

```
void sort(int* a, int n) {  
    for (int i = 1; i < n; ++i) {  
        int ith = i;  
        for (int j = i + 1; j <= n; ++j) {  
            if ( _____ ) {  
                ith = j;  
            }  
        }  
        int t = a[i];  
        a[i] = a[ith];  
        a[ith] = t;  
    }  
}
```

A $a[j] > a[j+1]$

B $a[j+1] > a[ith]$

C $a[j] < a[ith]$

D $a[j] > a[ith]$



Part 2

编程题

编程题

1、编写程序实现如下功能：

输入一个不超过50个字符的字符串且字符串中间无空格，把其中的字符按照逆序输出。

样例1： 输入：hello

输出：olleh

样例2： 输入：yeah

输出：haey

编程题

参考代码:

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string a;
    cin >> a;
    for(int i = a.size() - 1; i >= 0; i--)
    {
        cout << a[i];
    }
    return 0;
}
```

编程题

2、编写程序实现如下功能：

输入一个1-99999之间的正整数，求出它是几位数，并按逆序打印每个数。
(若输入值超出范围则返回false)

样例1： 输入：421

输出：3 124

样例2： 输入：123456

输出：false

编程题

参考代码:

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    int n, sum = 0, cnt = 0;
    cin >> n;
    if (n < 1 || n > 99999)
    {
        cout << "false";
        return 0;
    } else { while (n != 0) {
        sum = sum * 10 + n % 10;
        n /= 10;
        cnt++;
    }
    cout << cnt << " " << sum;
    return 0;
}
```

编程题

3、编写程序实现如下功能：

要求用户输入两个正整数 a, b ，其中 $a < b$ ，输出区间 $[a, b]$ 中所有的素数，以及素数的总个数。

说明：素数又称为质数，指的是能且只能被1和数字本身整除的数。

样例1：输入：10 20 输出：11 13 17 19 4

样例2：输入：5 17 输出：5 7 11 13 17 5

编程题

参考代码:

```
#include <iostream>
using namespace std;
int main()
{   int a, b, cnt = 0;
    cin >> a >> b;
    for (int i = a; i <= b; i++){
        if (i >= 2){
            bool flag = true;
            for (int j = 2; j <= i-1; j++){
                if (i % j == 0){
                    flag = false;
                    break;}
            }
            if (flag){
                cout << i << " ";
                cnt++;}
        }
    }
    cout << cnt;
    return 0;
}
```



Class is over