

2025第五次作业（选择题）

选择题

[刷新 ↻](#)

如无特殊说明，所有题目的编译选项都包含 `-std=c++11`

1. 【单选】下列关于 C++ 命名空间（namespace）的说法中，错误的是：
 - A. 命名空间可以嵌套，例如 `namespace A{namespace B{int i;}}`
 - B. 命名空间是开放的，可以随时添加新的成员
 - C. 命名空间不能在函数体内部定义
 - D. 使用 `using` 引入多个命名空间后，如果多个命名空间中存在同名函数，调用该函数时会在运行时出错
2. 【多选】关于 `vector` 迭代器失效，下列说法中正确的有：
 - A. 当调用 `vector::push_back` 向 `vector` 添加元素时，如果当前容量不足以容纳新元素，就会发生重新分配，从而使所有迭代器失效
 - B. 如果事先对 `vector` 调用了 `reserve`，保证了足够的容量，那么再进行 `push_back` 时，迭代器保持有效
 - C. 对 `vector` 调用 `erase(it)`，只会使被删除元素及其之后位置上所有元素的迭代器失效；在此之前的迭代器仍然有效
 - D. 对 `vector` 调用 `clear()` 清空容器，不会使任何迭代器失效
3. 【单选】下列关于简单容器类型及其相关函数，哪一项是不正确的？
 - A. `std::make_pair` 和 `std::make_tuple` 中，参数类型可以互不相同
 - B. `std::tie` 的形参和返回值均为左值引用
 - C. 使用 `std::get<i>(tup)` 获取元组 `tup` 中的第 `i` 个元素时，若 `i` 不是编译期常量，会导致编译错误
 - D. `tuple` 对象的长度只能在运行时才能确定
4. 【多选】以下代码展示了一些 `map` 类型上的操作，说法正确的是：

```

#include <iostream>
#include <map>
using namespace std;

void choiceA(map<string, int, greater<string>> age)
{
    age["Dave"] = age["Carol"] + 1;
    cout << age.size() << endl; // (1)
}

void choiceB(map<string, int, greater<string>> age)
{
    age.insert(make_pair("Alex", 5));
    cout << age["Alex"] << endl; // (2)
}

void choiceC(map<string, int, greater<string>> age)
{
    age["Alex"] = 5;
    cout << age["Alex"] << endl; // (3)
}

void choiceD(map<string, int, greater<string>> age)
{
    for (auto it = age.begin(); it != age.end(); it++)
    {
        cout << it->second << ' '; // (4a)
    }
    cout << endl; // (4b)
}

void choiceE(map<string, int, greater<string>> age)
{
    age.insert(make_pair("Carol", 5));
    age.erase(age.begin(), --age.end());
    cout << age.size() << endl;
}

int main()
{
    map<string, int, greater<string>> age;

    age["Alex"] = 2;
    age["Bob"] = 1;

    choiceA(age);
    choiceB(age);
    choiceC(age);
    choiceD(age);
    choiceE(age);
}

```

- A. choiceA() 中 (1) 的输出为 4
 - B. choiceB() 中 (2) 的输出为 5
 - C. choiceC() 中 (3) 的输出为 5
 - D. choiceD() 中 (4a)、(4b) 的输出为 2 1
5. 【单选】下列关于 Git 与 Markdown 的说法，哪一项是正确的？
- A. 在本地仓库根目录的 .gitignore 文件中加入 *.bin 后，新建的 a.bin 文件会被忽略，无法通过 git add a.bin 添加到版本库
 - B. 执行 git branch test 会创建并自动切换到 test 分支；执行 git branch -d test 会删除该分支
 - C. 克隆远程仓库后，如果在本地修改了文件但既未执行 git add 也未提交，则只能通过重新 git clone 来放弃这些本地修改
 - D. Markdown 中要同时实现 **加粗与删除线** 效果，应使用单波浪线包裹加粗语法：~**文本**~
6. 【多选】以下为在 linux 环境下的 script.sh，说法错误的是：

```

#!/bin/bash
filename=test
for par in "$@"; do
    mkdir output-$par || echo test$par > output-$par/test.txt
done

arr=(1 2 3)
arr[5]=10

```

- A. 在赋值 filename=test 时，为了美观可以写成 filename = test，脚本仍能运行
- B. 若当前目录下没有 output-0，执行 ./script.sh 0 时会执行 echo，将 test0 写入 output-0/test.txt
- C. 执行 ./script.sh 0 1 2 时，@为012，#为3

- D. 对 bash 数组执行 `arr[5]=10` 不会报错，会自动扩展数组
7. 【多选】下面关于 C++ 模板特化的说法，哪些是正确的？
- A. 函数模板不能进行部分特化
 - B. 类模板可以进行全部特化，也可以进行部分特化
 - C. 函数模板特化必须完全匹配指定的模板参数类型
 - D. 程序在运行时才会选择是使用基础函数模板还是特化的模板
8. 【多选】若在 `vector`、`list`、`set`、`map` 中选择，下列关于说法合适的有：
- A. 小明想维护一个排队购票系统，记录每个排队者的信息的同时，需要频繁从队尾加入新的排队者或从队头删去排队者的信息。针对这个需求，`list` 是最高效的选择
 - B. 小明想计算一篇文章中每个词出现的次数。针对这个需求，`map` 是合适的数据结构
 - C. 小明想将若干篇文章中所有出现的词整理成词表，要求每个词只能出现一次。针对这个需求，`set` 是合适的数据结构
 - D. 小明想建立一个排名 1 ~ 100 到学生姓名的映射。针对这个需求，`map` 是合适的选择，也可以选用 `vector` 更高效地实现

提交格式

请你提交一个文本文件，第*i*行是一个由A\B\C\D组成的字符串，代表第*i*题的答案。

若你不想提交第*i*题，请将第*i*行留空。

你提交的文本文件至少要有8行，且前10行必须由A\B\C\D组成，否则将被认为是无效提交。

你可以通过测试点的详细信息，看到评测对你提交文件的解析。若存在问题，请联系助教。

以下给出一个合法的提交答案的例子：

```
A
B
C
ACD
B
CD
C
C
```

评测器对第一行的解析结果如下：

```
Valid answer: ['A']
```

注意：作业截止之前本题的评测器只检查提交格式是否正确，不检查具体答案。只要提交格式正确就显示100分。每道题的具体答案会在作业截止之后重新评测，以最后一次提交的答案为准

语言和编译选项

#	名称	编译器	额外参数	代码长度限制
0	answer	cp		1048576 B

递交历史

#	状态	时间
343672	Accepted	2025-05-18 16:22:15

1

递交答案

选择文件

未选择文件

添加

提交

