

1405. Longest Happy String

A string s is called **happy** if it satisfies the following conditions:

- s only contains the letters 'a', 'b', and 'c'.
- s does not contain any of "aaa", "bbb", or "ccc" as a substring.
- s contains **at most** a occurrences of the letter 'a'.
- s contains **at most** b occurrences of the letter 'b'.
- s contains **at most** c occurrences of the letter 'c'.

Given three integers a , b , and c , return *the longest possible happy string*. If there are multiple longest happy strings, return *any of them*. If there is no such string, return *the empty string* "".

A **substring** is a contiguous sequence of characters within a string.

Example 1:

Input: $a = 1, b = 1, c = 7$

Output: "ccacccbcc"

Explanation: "ccbccacc" would also be a correct answer.

Example 2:

Input: $a = 7, b = 1, c = 0$

Output: "aabaa"

Explanation: It is the only correct answer in this case.

Constraints:

- $0 \leq a, b, c \leq 100$
- $a + b + c > 0$

1405. Longest Happy String

```
typedef std::pair<char,int> ci;
```

```
typedef std::vector<ci> vci;
```

```
class Solution{
```

```
public:
```

```
    string longestDiverseString(int a, int b, int c){
```

```
        std::priority_queue<ci,vci> max_heap;
```

```
        if(a>0) max_heap.push({a,'a'});
```

```
        if(b>0) max_heap.push({b,'b'});
```

```
        if(c>0) max_heap.push({c,'c'});
```

```
        std::string ans="";
```

```
        while(!max_heap.empty()){
```

```
            auto [f1,c1]=max_heap.top();
```

```
            max_heap.pop();
```

```
            if(ans.size()>=2 && ans[ans.size()-1]==c1 && ans[ans.size()-2]==c1){
```

```
                if(max_heap.empty()) break;
```

```
                auto [f2,c2]=max_heap.top();
```

```
                max_heap.pop();
```

```
                ans+=c2;
```

```
                f2--;
```

```
                if(f2>0) max_heap.push({f2,c2});
```

```
            }
```

```
            else{
```

```
                ans+=c1;
```

```
                f1--;
```

```
            }
```

```
            if(f1>0) max_heap.push({f1,c1});
```

```
        }
```

```
        return ans;
```

```
    }
```

```
};
```