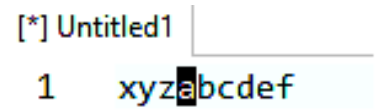
Insert Delete

Tony Wong

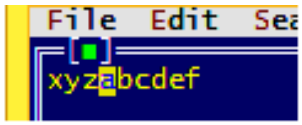
2015-01-17

Task Description

- In Overtype mode, characters will be replaced as you type.



Dev-C++



Free Pascal

- Given: original text, key presses to simulate
- Output the final text

Pretest #3	
xelanoop IxelaIgnutDDDD	xelagnut

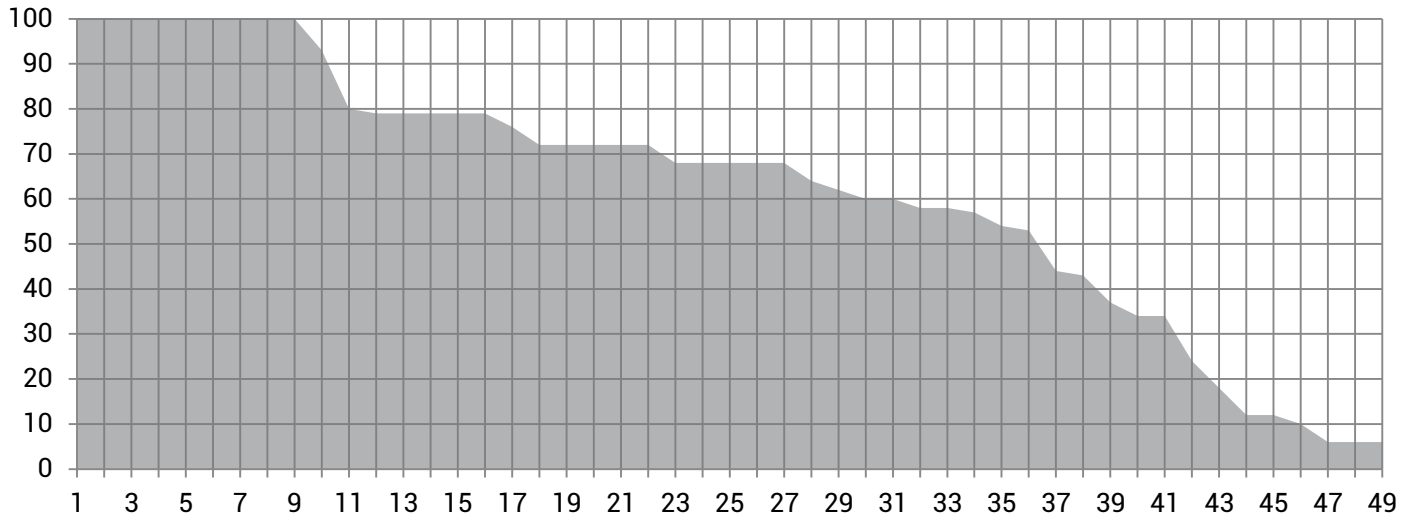
Preface

- Intended to be an easy task because String processing is inherently difficult
- Test contestants their familiarity with the programming language
- Many ways to implement
- Results are better than expected

Insert Delete

Statistics

- Attempt: 67 No attempt: 11
- 100: 9 0: 18 Avg: 45.61 Median: 57
- First solved by Lynn Shung Hei @ 25m 15s



Observation 1

- Any new characters cannot be replaced
- The characters will also be in the same order

Sample #1	
heat fDiDnDaDl	final

Observation 2

- The new characters will be in the front of the old characters

Sample #2	
senior Iju	junior

Insert Delete

Observation 3

- Number of characters removed from original text is equal to
- Number of key presses during Overtype mode + Number of Delete presses

Sample #3	
remember 6thDdIeIcD	6thdecember

Therefore

- Base on Observation 1 & 2, we can output the new characters immediately.
- Base on Observation 3, calculate the correct number of the remaining characters.

Pretest #3	
xelanoop IxelaIgnutDDDD	xelagnut

Official Solution

```cpp
#include <cstdio>
#include <cstring>
char a[100001], b[100001];
int main() {
  int x, y;
  scanf("%s %s", a, b);
  x = strlen(a);
  y = strlen(b);
  int overtype = 0;
  int deleted = 0;
  for (int i = 0; i < y; i++) {
    if (b[i] == 'D') {
      deleted++;
    } else if (b[i] == 'I') {
      overtype = 1 - overtype;
    } else {
      printf("%c", b[i]);
      deleted += overtype;
    }
  }
  if (deleted < x) {
    printf("%s", a + deleted);
  }
  printf("\n");
  return 0;
}
```

Official Solution

- It is rare that the Pascal solution is shorter than the C++ one.

```pascal
var
  a, b: ansistring;
  i, y, overtype, deleted: longint;
begin
  readln(a);
  readln(b);
  y := length(b);
  overtype := 0;
  deleted := 0;
  for i := 1 to y do
  begin
    if (b[i] = 'D') then
      inc(deleted)
    else if (b[i] = 'I') then
      overtype := 1 - overtype
    else
    begin
      write(b[i]);
      deleted := deleted + overtype;
    end;
  end;
  delete(a, 1, deleted);
  writeln(a);
end.
```

Common mistakes

- Depending on implementation, you may have to check whether there are characters remaining before deleting / overwriting (Runtime error)

System Test #5	
1234 DDDDDDDDDDD...	
System Test #7	
1234567890 opqIsItuIooooooooo...	opqstuooo...

Common mistakes

- Solutions that uses strcpy (copy, delete) heavily may exceed time limit.
- If you append characters to the original string, the size has to be 100000 x 2 + 1 = 200001 (Runtime error). See student's solution below:

```
1  #include <cstdio>
2  #include <cstring>
3
4  char output[200001];
5  char cmd[100001];
6  int cursor=0;
7  bool mode=true;//true: insert, false:overwrite
8  int cmdReadHead=0;
```

Tips

- DO NOT USE strlen(s) in loops. It is slow!
- Instead, save the length into a variable if possible

```
 8    y = strlen(b);
 9    int overtype = 0;
10    int deleted = 0;
11    for (int i = 0; i < y; i++) {
```