**A**

**C**

**M**

**模**

**板**

**-------lsy**

# 赛场注意事项

## 1. 测试机器

### 1.1 鼠标键盘测试

### 1.2 软件测试是否可用

### 1.3 内存大小

最大数组

栈大小

扩栈测试

## 2. 基本设定

### codeblocks

terminal： gnome-terminal -t $TITLE -x 在codeblocks --> setting --> 环境变量

### eclipse

Eclipse中默认是输入"."后出现自动提示，用于类成员的自动提示，可是有时候我们希望它能在我们输入类的首字母后就出现自动提示，可以节省大量的输入时间（虽然按alt + /会出现提示，但还是要多按一次按键，太麻烦了）。    从Window -> preferences -> Java -> Editor -> Content assist -> Auto-Activation下，我们可以在"."号后面加入我们需要自动提示的首字幕，比如"ahiz"。    然后我们回到Eclipse的开发环境，输入"a"，提示就出现了。但是我们可以发现，这个Auto-Activation下的输入框里最多只能输入5个字母，也许是Eclipse的开发人员担心我们输入的太多会影响性能，但计算机的性能不用白不用，所以我们要打破这个限制。其实上面都是铺垫，制造一下气氛，以显得我们下面要做的事情很牛似的，其实不然，一切都很简单。嘿嘿 :)

在"."后面随便输入几个字符，比如"abij"，然后回到开发环境，File -> export -> general -> preferences -> 选一个地方保存你的首选项，比如C:"a.epf用任何文本编辑器打开a.epf，查找字符串“abij”，找到以后，替换成“abcdefghijklmnopqrstuvwxyz”，总之就是你想怎样就怎样！！然后回到Eclipse，File -> import -> general -> preferences -> 导入刚才的a.epf文件。此时你会发现输入任何字幕都可以得到自动提示了。爽！！！最后：自动提示弹出的时间最好改成100毫秒以下，这样会比较爽一点，不然你都完事了，自动提示才弹出来:)，不过也要看机器性能。

FileWriter fileWriter=new FileWriter("c:\\Result.txt");

int [] a=new int[]{11112,222,333,444,555,666};

for (int i = 0; i < a.length; i++) {

fileWriter.write(String.valueOf(a[i])+" ");

}

## 3. ASCII

0 NUL(null) 空字符

1 SOH(start of headline) 标题开始

2 STX (start of text) 正文开始

3 [ETX](http://baike.baidu.com/view/13854.htm)(end of text) 正文结束

4 EOT (end of transmission) 传输结束

5 ENQ (enquiry) 请求

6 ACK (acknowledge) 收到通知

7 BEL (bell) 响铃

8 BS (backspace) 退格

9 HT (horizontal tab) 水平制表符

10 LF (NL line feed, new line) 换行键

11 VT (vertical tab) 垂直制表符

12 FF (NP form feed, new page) 换页键

13 CR (carriage return) 回车键

14 SO (shift out) 不用切换

15 SI (shift in) 启用切换

16 DLE (data link escape) 数据链路转义

17 DC1 (device control 1) 设备控制1

18 DC2 (device control 2) 设备控制2

19 DC3 (device control 3) 设备控制3

20 DC4 (device control 4) 设备控制4

21 NAK (negative acknowledge) 拒绝接收

22 SYN (synchronous idle) 同步空闲

23 ETB (end of trans. block) 传输块结束

24 CAN (cancel) 取消

25 EM (end of medium) 介质中断

26 SUB (substitute) 替补

27 ESC (escape) 换码(溢出)

28 FS (file separator) 文件分割符

29 GS (group separator) 分组符

30 ‑ RS (record separator) 记录分离符

31 ­ US (unit separator) 单元分隔符

32 space 空格

33 !

34 "

35 #

36 $

37 %

38 &

39 '

40 (

41 )

42 \*

43 +

44 ,

45 -

46 .

47 /

48 0

49 1

50 2

51 3

52 4

53 5

54 6

55 7

56 8

57 9

58 :

59 ;

60 <

61 =

62 >

63 ?

64 @

65 A

66 B

67 C

68 D

69 E

70 F

71 G

72 H

73 I

74 J

75 K

76 L

77 M

78 N

79 O

80 P

81 Q

82 R

83 S

84 T

85 U

86 V

87 W

88 X

89 Y

90 Z

91 [

92 \

93 ]

94 ^

95 \_

96 `

97 a

98 b

99 c

100 d

101 e

102 f

103 g

104 h

105 i

106 j

107 k

108 l

109 m

110 n

111 o

112 p

113 q

114 r

115 s

116 t

117 u

118 v

119 w

120 x

121 y

122 z

123 {

124 |

125 }

126 ~

127 DEL（delete 删除）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bin | Dec | Hex | 缩写/字符 | 解释 | |
| 0000 0000 | 0 | 00 | NUL(null) | 空字符 | |
| 0000 0001 | 1 | 01 | SOH(start of headline) | 标题开始 | |
| 0000 0010 | 2 | 02 | STX (start of text) | 正文开始 | |
| 0000 0011 | 3 | 03 | ETX (end of text) | 正文结束 | |
| 0000 0100 | 4 | 04 | EOT (end of transmission) | 传输结束 | |
| 0000 0101 | 5 | 05 | ENQ (enquiry) | 请求 | |
| 0000 0110 | 6 | 06 | ACK (acknowledge) | 收到通知 | |
| 0000 0111 | 7 | 07 | BEL (bell) | 响铃 | |
| 0000 1000 | 8 | 08 | BS (backspace) | 退格 | |
| 0000 1001 | 9 | 09 | HT (horizontal tab) | 水平制表符 | |
| 0000 1010 | 10 | 0A | LF (NL line feed, new line) | 换行键 | |
| 0000 1011 | 11 | 0B | VT (vertical tab) | 垂直制表符 | |
| 0000 1100 | 12 | 0C | FF (NP form feed, new page) | 换页键 | |
| 0000 1101 | 13 | 0D | CR (carriage return) | 回车键 | |
| 0000 1110 | 14 | 0E | SO (shift out) | 不用切换 | |
| 0000 1111 | 15 | 0F | SI (shift in) | 启用切换 | |
| 0001 0000 | 16 | 10 | DLE (data link escape) | 数据链路转义 | |
| 0001 0001 | 17 | 11 | DC1 (device control 1) | 设备控制1 | |
| 0001 0010 | 18 | 12 | DC2 (device control 2) | 设备控制2 | |
| 0001 0011 | 19 | 13 | DC3 (device control 3) | 设备控制3 | |
| 0001 0100 | 20 | 14 | DC4 (device control 4) | 设备控制4 | |
| 0001 0101 | 21 | 15 | NAK (negative acknowledge) | 拒绝接收 | |
| 0001 0110 | 22 | 16 | SYN (synchronous idle) | 同步空闲 | |
| 0001 0111 | 23 | 17 | ETB (end of trans. block) | 传输块结束 | |
| 0001 1000 | 24 | 18 | CAN (cancel) | 取消 | |
| 0001 1001 | 25 | 19 | EM (end of medium) | 介质中断 | |
| 0001 1010 | 26 | 1A | SUB (substitute) | 替补 | |
| 0001 1011 | 27 | 1B | ESC (escape) | 换码(溢出) | |
| 0001 1100 | 28 | 1C | FS (file separator) | 文件分割符 | |
| 0001 1101 | 29 | 1D | GS (group separator) | 分组符 | |
| 0001 1110 | 30 | 1E | RS (record separator) | 记录分离符 | |
| 0001 1111 | 31 | 1F | US (unit separator) | 单元分隔符 | |
| 0010 0000 | 32 | 20 | (space) | 空格 | |
| 0010 0001 | 33 | 21 | ! |  | |
| 0010 0010 | 34 | 22 | " |  | |
| 0010 0011 | 35 | 23 | # |  | |
| 0010 0100 | 36 | 24 | $ |  | |
| 0010 0101 | 37 | 25 | % |  | |
| 0010 0110 | 38 | 26 | & |  | |
| 0010 0111 | 39 | 27 | ' |  | |
| 0010 1000 | 40 | 28 | ( |  | |
| 0010 1001 | 41 | 29 | ) |  | |
| 0010 1010 | 42 | 2A | \* |  | |
| 0010 1011 | 43 | 2B | + |  | |
| 0010 1100 | 44 | 2C | , |  | |
| 0010 1101 | 45 | 2D | - |  | |
| 0010 1110 | 46 | 2E | . |  | |
| 00101111 | 47 | 2F | / |  | |
| 00110000 | 48 | 30 | 0 |  | |
| 00110001 | 49 | 31 | 1 |  |  |
| 00110010 | 50 | 32 | 2 |  |  |
| 00110011 | 51 | 33 | 3 |  |  |
| 00110100 | 52 | 34 | 4 |  |  |
| 00110101 | 53 | 35 | 5 |  |  |
| 00110110 | 54 | 36 | 6 |  |  |
| 00110111 | 55 | 37 | 7 |  |  |
| 00111000 | 56 | 38 | 8 |  |  |
| 00111001 | 57 | 39 | 9 |  |  |
| 00111010 | 58 | 3A | : |  |  |
| 00111011 | 59 | 3B | ; |  |  |
| 00111100 | 60 | 3C | < |  |  |
| 00111101 | 61 | 3D | = |  |  |
| 00111110 | 62 | 3E | > |  |  |
| 00111111 | 63 | 3F | ? |  |  |
| 01000000 | 64 | 40 | @ |  |  |
| 01000001 | 65 | 41 | A |  |  |
| 01000010 | 66 | 42 | B |  |  |
| 01000011 | 67 | 43 | C |  |  |
| 01000100 | 68 | 44 | D |  |  |
| 01000101 | 69 | 45 | E |  |  |
| 01000110 | 70 | 46 | F |  |  |
| 01000111 | 71 | 47 | G |  |  |
| 01001000 | 72 | 48 | H |  |  |
| 01001001 | 73 | 49 | I |  |  |
| 01001010 | 74 | 4A | J |  |  |
| 01001011 | 75 | 4B | K |  |  |
| 01001100 | 76 | 4C | L |  |  |
| 01001101 | 77 | 4D | M |  |  |
| 01001110 | 78 | 4E | N |  |  |
| 01001111 | 79 | 4F | O |  |  |
| 01010000 | 80 | 50 | P |  |  |
| 01010001 | 81 | 51 | Q |  |  |
| 01010010 | 82 | 52 | R |  |  |
| 01010011 | 83 | 53 | S |  |  |
| 01010100 | 84 | 54 | T |  |  |
| 01010101 | 85 | 55 | U |  |  |
| 01010110 | 86 | 56 | V |  |  |
| 01010111 | 87 | 57 | W |  |  |
| 01011000 | 88 | 58 | X |  |  |
| 01011001 | 89 | 59 | Y |  |  |
| 01011010 | 90 | 5A | Z |  |  |
| 01011011 | 91 | 5B | [ |  |  |
| 01011100 | 92 | 5C | \ |  |  |
| 01011101 | 93 | 5D | ] |  |  |
| 01011110 | 94 | 5E | ^ |  |  |
| 01011111 | 95 | 5F | \_ |  |  |
| 01100000 | 96 | 60 | ` |  |  |
| 01100001 | 97 | 61 | a |  |  |
| 01100010 | 98 | 62 | b |  |  |
| 01100011 | 99 | 63 | c |  |  |
| 01100100 | 100 | 64 | d |  |  |
| 01100101 | 101 | 65 | e |  |  |
| 01100110 | 102 | 66 | f |  |  |
| 01100111 | 103 | 67 | g |  |  |
| 01101000 | 104 | 68 | h |  |  |
| 01101001 | 105 | 69 | i |  |  |
| 01101010 | 106 | 6A | j |  |  |
| 01101011 | 107 | 6B | k |  |  |
| 01101100 | 108 | 6C | l |  |  |
| 01101101 | 109 | 6D | m |  |  |
| 01101110 | 110 | 6E | n |  |  |
| 01101111 | 111 | 6F | o |  |  |
| 01110000 | 112 | 70 | p |  |  |
| 01110001 | 113 | 71 | q |  |  |
| 01110010 | 114 | 72 | r |  |  |
| 01110011 | 115 | 73 | s |  |  |
| 01110100 | 116 | 74 | t |  |  |
| 01110101 | 117 | 75 | u |  |  |
| 01110110 | 118 | 76 | v |  |  |
| 01110111 | 119 | 77 | w |  |  |
| 01111000 | 120 | 78 | x |  |  |
| 01111001 | 121 | 79 | y |  |  |
| 01111010 | 122 | 7A | z |  |  |
| 01111011 | 123 | 7B | { |  |  |
| 01111100 | 124 | 7C | | |  |  |
| 01111101 | 125 | 7D | } |  |  |
| 01111110 | 126 | 7E | ~ |  |  |
| 01111111 | 127 | 7F | DEL (delete) | 删除 |  |

#### 5.DLX

/\* dancing link

\* 精确覆盖问题

\* 可以添加迭代加深优化：

\* 1）枚举深度h；

\* 2）若当前深度+predeep > h return false；

\*

int predeep() {

bool vis[Maxm];

memset(vis, 0, sizeof(vis));

int ret = 0;

for (Node \*p = head->R; p != head; p = p->R)

if (!vis[p->col]) {

ret ++ ;

vis[p->col] ++ ;

for (Node \*q = p->D; q != p; q = p->D)

for (Node \*r = q->R; r != q; r = r->R)

vis[r->col] = true;

}

return ret;

}

\* \*/

#define Maxn 1010

#define Maxm 1010

struct Node {

Node \*L, \*R, \*U, \*D;

int col, row;

} \*head, \*row[Maxn], \*col[Maxm], node[Maxn \* Maxm];

int colsum[Maxm], cnt;

void init(int mat[][Maxm], int n, int m) {

cnt = 0;

memset(colsum, 0, sizeof(colsum));

head = &node[cnt ++ ];

for (int i = 1; i <= n; i ++ )

row[i] = &node[cnt ++ ];

for (int j = 1; j <= m; j ++ )

col[j] = &node[cnt ++ ];

head->D = row[1], row[1]->U = head;

head->R = col[1], col[1]->L = head;

head->U = row[n], row[n]->D = head;

head->L = col[m], col[m]->R = head;

head->row = head->col = 0;

for (int i = 1; i <= n; i ++ ) {

if (i != n) row[i]->D = row[i + 1];

if (i != 1) row[i]->U = row[i - 1];

row[i]->L = row[i]->R = row[i];

row[i]->row = i, row[i]->col = 0;

}

for (int i = 1; i <= m; i ++ ) {

if (i != m) col[i]->R = col[i + 1];

if (i != 1) col[i]->L = col[i - 1];

col[i]->U = col[i]->D = col[i];

col[i]->col = i, col[i]->row = 0;

}

for (int i = n; i > 0; i -- )

for (int j = m; j > 0; j -- )

if (mat[i][j]) {

Node \*p = &node[cnt ++ ];

p->R = row[i]->R, row[i]->R->L = p;

p->L = row[i], row[i]->R = p;

p->D = col[j]->D, col[j]->D->U = p;

p->U = col[j], col[j]->D = p;

p->row = i;

p->col = j;

colsum[j] ++ ;

}

}

/\*多重覆盖只需删除列，无需对应行删除

void remove(Node \*c) {

for (Node \*p = c->D; p != c; p = p->D) {

p->L->R = p->R;

p->R->L = p->L;

}

}

\*/

void remove(Node \*c) {

c->L->R = c->R;

c->R->L = c->L;

for (Node \*p = c->D; p != c; p = p->D) {

for (Node \*q = p->R; q != p; q = q->R) {

q->U->D = q->D;

q->D->U = q->U;

colsum[q->col] -- ;

}

}

}

void resume(Node \*c) {

for (Node \*p = c->U; p != c; p = p->U) {

for (Node \*q = p->L; q != p; q = q->L) {

q->U->D = q;

q->D->U = q;

colsum[q->col] ++ ;

}

}

col[c->col]->L->R = col[c->col];

col[c->col]->R->L = col[c->col];

}

int ans[Maxm];

int dfs(int deep) {

if (head->R == head) return deep;

Node \*p, \*q = head->R;

for (p = head->R; p != head; p = p->R)

if (colsum[p->col] < colsum[q->col])

q = p;

remove(q);

for (p = q->D; p != q; p = p->D) {

for (Node\* r = p->R; r != p; r = r->R)

if (r->col != 0)

remove(col[r->col]);

/\*--------可修改区域-----------\*/

ans[deep] = p->row;

/\*-----------------------------\*/

int sta = dfs(deep + 1);

if (sta != -1) return sta;

for (Node\* r = p->L; r != p; r = r->L)

if (r->col != 0)

resume(col[r->col]);

}

resume(q);

return -1;

}

//#pragma comment(linker,"/STACK:102400000,102400000")

#include <cstdio>

#include <cstring>

#include <cstdlib>

#include <cmath>

#include <cctype>

#include <iostream>

#include <sstream>

#include <algorithm>

#include <string>

#include <vector>

#include <set>

#include <map>

#include <stack>

#include <queue>

#define N 10100

#define M 200100

#define LEN 100100

#define INF (1 << 30)

#define eps 1e-10

#define PB push\_back

#define A first

#define B second

using namespace std;

typedef long long LL;