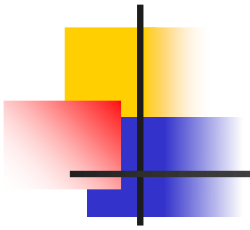
Functions with Pass-by-Reference

Lab 10: Guessing a Password

Rung-Bin Lin

International Bachelor Program in Informatics
Yuan Ze University

12/10/2022



Purposes of the Lab

- **More on**
  - ✓ **Reference type**
  - ✓ **Pass by value & pass by reference**

Reference Type

```
int main( ) {
int x =1;
int &y = x;  // Reference type should be initialized
int xx = 3;
int &zz = xx;
int anAry[] = {1, 2, 3, 4}; // declare
int aFunc1(int &);
int aFunc2(int &aRef);
int aFrnc22(int &aRef);
void aFun3(int);
int &aFunc4(int &aRef);
int& aFunc5(int anAry[], int);
int& aFunc6(int anAry[], int &);

cout << aFunc1(x); // 4 printed; x is 1
cout << aFunc2(xx); // 4 printed; xx is 4
cout << aFunc22(xx); // 4 printed; xx is 5
cout << aFunc3(y); // not a legal statement
cout << aFunc4(xx); // 6 printed, xx is 6
cout << aFunc5(anAry, 4); //10 printed
cout << aFunc6(anAry, 4); // not a legal
statement
return 0;
```

```
int aFunc1(int &a){ return a+3; }

int aFunc2(int &aRef) {return ++aRef};

Int aFunc22(int &aRef) (return aRef++);

void aFun3(int x) {x++;}

Int &aFunc4(int &aRef) {aRef++; return aRef;}

Int& aFunc5(int anAry[], int inx){
anAry[2] = 10;
return anAry[2] ;}

Int& aFunc6(int anAry[], int & inx) {
anAry[2] =10;
inx++;
return anAry[2]; }
```

Lab 10: Guess a Password

- Write a program that will guess a password as follows:
  - **You are given a function string generatePassWd(int&) to generate a password that contains at most four lower-case letters, for example "abcd". The function should return a password. The parameter should return the length of the password.**
  - **Read from a keyboard a string which is the guess you made. Print out "Too high" if the string read from the keyboard is greater than passWd or print out "Too low" if it is smaller than passWd. Strings are compared in terms of their lexicographic order. For example, a < b, aa < ab, abc < abca, etc.**
  - **You should continue to read strings from the keyboard until you guess the password right. That is, the password read from the keyboard is the same as passWd.**
  - **If a right guess is made, print "Bravo, you guess it right!". Moreover, if the number of guesses you made for a right guess is smaller than or eqial to $X=\lceil \log_2 26^{len} \rceil+2$, then print out "You know the secret!", where len is the length of a password and ⌈ ⌉ is the *ceiling* function (page 194). Otherwise, print out "You should be able to do better." Here, you should use log2() function rather than log() or log10().**

string generatePassWd(int &)

```
string generatePassWd(int &passLen){
    string tempStr=“”;
    srand(time(0));
    passLen = rand()%4+1;
    for (int i=0; i<passLen; i++){
        tempStr = tempStr + “a”;
        tempStr[i] = 'a' + rand()% 26;
    }
     return tempStr;
}
```

Scoring

- Your score will be calculated as follows:
- **min (100, (1 − 0.5 ∗ (#ofGuess-X)/X)*100)**
  - For example, if X = 20, and #ofGuess = 22, then the score will be
    min(100, (1-0.5*(22-20)/20)*100) = min(100, 95) =95.

Requirements (1)

- **Write a function**

  status guess(string, string);

  where status is an enumeration type:

  enum status {TH, TL, RT}; // TH: too high; TL: too low; RT: right

  The function should return a guess result, TH, TL, or RT. The first parameter is a guess. The second parameter is the string we would like to guess.

- **The main() function should have calls to this function as follows:**

```
Int main()
{
  string aGuess;  // A guess read from keyboard
  string passWd; // passWd we would like to guess
  int passLen;    // passWd length
        …
   guess(aGuess, passWD);
        …
}
```

Requirements (2)

- **After a right guess is made, your program should ask whether to play the game again by presenting a prompt message “Play the game again (Y or y for yes): “. Otherwise, the program terminates.**

Example of Input & Output

```
Guessing a password at most having four lower-case letters. My guess is as follows:
1-st guess = oasx
Bravo, you guess it right!
You know the secret!
Play the game again (Y or y for yes): y
1-st guess = a
Too low. Try again.
2-nd guess = aa
Too low. Try again.
3-rd guess = aaa
Too low. Try again.
4-th guess = aaaa
Too low. Try again.
5-th guess = mmmm
Too high. Try again.
6-th guess = jjjj
Too low. Try again.
7-th guess = kkkk
Too high. Try again.
8-th guess = jjzz
Too low. Try again.
9-th guess = jzzz
Too low. Try again.
10-th guess = kkaa
Too high. Try again.
11-th guess = kazz
Too high. Try again.
12-th guess = kaaz
Too low. Try again.
13-th guess = kfaa
Too high. Try again.
14-th guess = kcaa
Too high. Try again.
15-th guess = kbaa
Too high. Try again.
16-th guess = kazz
Too high. Try again.
17-th guess = kall
Too low. Try again.
18-th guess = kaha
Too low. Try again.
19-th guess = kaja
Too low. Try again.
20-th guess = kaka
Too low. Try again.
21-th guess = kakz
Too low. Try again.
22-th guess = kala
Too low. Try again.
23-th guess = kalz
Too low. Try again.
24-th guess = kamz
Too low. Try again.
25-th guess = kazz
Too high. Try again.
26-th guess = kaoz
Too low. Try again.
27-th guess = kavz
Too high. Try again.
28-th guess = kaqz
Too low. Try again.
29-th guess = karz
Too low. Try again.
30-th guess = katz
Too high. Try again.
31-th guess = kasz
Too high. Try again.
32-th guess = kasa
Too low. Try again.
33-th guess = kasl
Too low. Try again.
34-th guess = kasp
Too low. Try again.
35-th guess = kasu
Too low. Try again.
36-th guess = kasw
Too low. Try again.
37-th guess = kasy
Too high. Try again.
38-th guess = kasx
Bravo, you guess it right!
You should be able to do better.
Play the game again (Y or y for yes):
```

Note: You should have 1-st, 2-nd, 3-rd, 4-th, …, 10-th, 11-th, 12-th, 13-th, …, 20-th, 21-th, ….

- The above is not a good way of guessing the password.

Another Example

```
Guessing a password at most having four lower-case letters. My guess is as follows:
1-st guess = zzz
Too low. Try again.
2-nd guess = mmmm
Too low. Try again.
3-rd guess = tttt
Too high. Try again.
4-th guess = pppp
Too low. Try again.
5-th guess = rrrr
Too low. Try again.
6-th guess = ssss
Too low. Try again.
7-th guess = szzz
Too high. Try again.
8-th guess = smmm
Too low. Try again.
9-th guess = sttt
Too low. Try again.
10-th guess = swww
Too high. Try again.
11-th guess = svvv
Too high. Try again.
12-th guess = svzz
Too high. Try again.
13-th guess = svaa
Too low. Try again.
14-th guess = svmm
Too low. Try again.
15-th guess = svtt
Too high. Try again.
16-th guess = svqq
Too low. Try again.
17-th guess = svss
Too high. Try again.
18-th guess = svrr
Too low. Try again.
19-th guess = svsa
Too low. Try again.
20-th guess = svsl
Too high. Try again.
21-th guess = svsf
Too low. Try again.
22-th guess = svsi
Too low. Try again.
23-th guess = svsk
Too high. Try again.
24-th guess = svsj
Bravo, you guess it right!
You should be able to do better.
```

Bonus: Automatic Guessing

- A bonus of 20 points.
- Write a program to make an automatic guess. That is, repeat making a guess and receiving a response until a right guess is made.
- For this part, you can change **status guess(string, string)** freely or replace this function with another function.

An Example of Optimal Automatic Guessing

```
Guessing a password having at most four lower-case letters.

The password to be guessed: vw
1-st guess = mmmm
Too high.
2-nd guess = fsst
Too high.
3-rd guess = civw
Too high.
4-th guess = aqxl
Too high.
5-th guess = uyf
Too high.
6-th guess = jyp
Too high.
7-th guess = elu
Too high.
8-th guess = bsk
Too high.
9-th guess = ais
Too high.
10-th guess = qw
Too low.
11-th guess = zu
Too high.
12-th guess = vi
Too low.
13-th guess = xo
Too high.
14-th guess = wl
Too high.
15-th guess = vw
Bravo, you guess it right!
You know the secret!
Play the game again (Y or y for yes): y
```

```
The password to be guessed: ncfd
1-st guess = mmmm
Too low.
2-nd guess = tggf
Too high.
3-rd guess = pwji
Too high.
4-th guess = oekx
Too high.
5-th guess = nilr
Too high.
6-th guess = mxmb
Too low.
7-th guess = ncyw
Too high.
8-th guess = nafl
Too low.
9-th guess = nbpd
Too low.
10-th guess = ncgz
Too high.
11-th guess = nbyb
Too low.
12-th guess = nccn
Too low.
13-th guess = ncet
Too low.
14-th guess = ncfw
Too high.
15-th guess = ncfh
Too high.
16-th guess = ncfa
Too low.
17-th guess = ncfd
Bravo, you guess it right!
You know the secret!
```