8.1

a) address

b) 0, NULL, an address

c) 0

8.2

a) False. It can also be applied to variaties.

b) False. A pointer of type void \* cannot be dereferenced.

c) False. Any type of pointer can be assigned to a void \* pointer.

8.3

a) double numbers[SIZE] = {0.0,1.1,2.2,3.3,4.4,5.5,6.6,7.7,8.8,9.9};

b) double \*nPtr;

c) for(int i = 0;i< SIZE;++i)

cout << setprecision(1) << numbers[i];

d) nPtr = numbers;

nPtr = &numbers[0];

e) for(int i = 0;i < SIZE;++i)

cout << \*(nPtr+i);

f) for(int i = 0;i < SIZE;++i)

cout << \*(numbers+i);

g) for(int i = 0;i < SIZE;++i)

cout << nPtr[i];

h) numbers[3], \*(numbers+3), nPtr[3], \*(nPtr+3)

i) 1002564, 8.8.

j) 1002508, 1.1.

8.4

a) double \*fPtr;

b) fPtr = &number1;

c) cout << \*fPtr;

d) number2 = \*fPtr;

e) cout << number2;

f) cout << &number1;

g) cout << fPtr; Yes.

8.5

a) void exchange (double \*x, double \*y);

b) void exchange (double \*, double \*);

c) int evaluate(int x, int (\*polyPtr) (int));

d) int evaluate(int, int (\*)(int));

e) char vowel[] = “AEIOU”;

char vowel[] = {A,E,I,O,U,\0};

8.6

a) zPtr未初始化。

b) number实际赋的值为数列z第0个元素的地址，应为number = \*zPtr;

c) number = zPtr[2]; OR number = \*(zPtr + 2); OR number = z[2]; OR number = \*(z + 2);

d) 多循环了一次，应将i <= 5 改为 i < 5;

e) 间接引用了void类型值。应该为number = static\_cast<int>(\*sPtr);

f) 数组名是const类型的指针，指向位置不可改变，不能自增。

8.7

a) True.

b) True.

8.8  
a) unsigned int values[SIZE] = {2,4,6,8,10};

b) unsigned int \*vPtr;

c) for(int i = 0;i < SIZE;++i)

cout << values[i];

d) vPtr = values;

vPtr = &values[0];

e) for(int i = 0;i < SIZE;++i)

cout << \*(vPtr + i);

f) for(int i = 0;i < SIZE;++i)

cout << \*(values + i);

g) for(int i = 0;i < SIZE;++i)

cout << vPtr[i];

h) values[4], \*(value + 4), vPrt[4], \*(vPtr + 4),

i) 1002512, 8

j) 1002500, 2

8.9

a) long \*longPtr;

b) longPtr = &value1;

c) cout << \*longPtr;

d) value2 = \*longPtr;

e) cout << value2;

f) cour << &value1;

g) cout << longPtr; Yes.

8.10

a) void zero(long bigIntegers[]);

b) void zero(long []);

c) int add1AndSum(int oneTooSmall);

d) int add1AndSum(int);

8.11

a) number未被初始化。改正：number = 00000000;

b) 不能用double类型的指针给int类型的指针赋值

c) 不能将变量的值赋给指针。改正：int \*x, \*y;

d) s是数组名，指向的位置一定，不能自增。

改正：for(int i = 0;\*(s + i) != ‘\0’;++i)

cout << \*(s + i) << ‘ ‘;

e) 不能用其他类型的指针给void类型指针赋值。

f) 输出了x的地址。改正：cout << \*xPtr << endl;

8.13 在第一个字符串的空余部分加上第二个字符串，并在第80个字符处截断。

8.14 统计字符串中字符数。

**华南理工大学软件学院**

**高级语言程序设计报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓名** | 叶子繁 | **学号** | 201730684427 | **开题时间** | 2017.12.14 |
| **班级** | 软件工程（4）班 | | | **任课教师** | 金龙存 |
| **设计题目** | 8.12 | | | | |
| **一．题目功能描述**  龟兔赛跑。  **二．设计思路**  使用int的数据类型；  使用数组；  使用指针对数组内的数据进行修改；  使用rand和srand进行随机；  使用cin和cout进行数据的输入输出；  **三．实现代码**  #include<iostream>  #include<stdlib.h>  #include<ctime>  #include<iomanip>  using namespace std;  void moveTortoise(int \*);  void moveHare(int \*);  void clearLine(int \*,int \*, char \*);  bool moveLine(int \*,int \*,char \*);  int main()  {  srand(time(0));  cout << "BANG !!!!!\nAND THEY ARE OFF !!!!!" << endl;  char line[70];  for(int i = 0;i<70;++i)  line[i] = ' ';  int torPos = 0;  int harePos = 0;  int \*tPtr,\*hPtr;  char \*lPtr = line;  tPtr = &torPos;  hPtr = &harePos;  while(torPos<70&&harePos<70)  {  clearLine(tPtr,hPtr,lPtr);  moveHare(hPtr);  moveTortoise(tPtr);  if(moveLine(tPtr,hPtr,lPtr))  {  for(int i = 0;i<69;++i)  cout << line[i];  cout << line[69] << endl;  }  else  cout<< "OUCH!!!" << endl;  }  if(torPos>=70&&harePos<70)  cout << "TORTOISE WINS!!! YAY!!!" << endl;  else if(torPos<70&&harePos>=70)  cout << "Hare wins. Yuch." << endl;  else  cout << "It's a tie." << endl;  }  void moveTortoise(int \*tPtr)  {  switch (rand()%10)  {  case 0:  \*tPtr +=3;  break;  case 1:  \*tPtr +=3;  break;  case 2:  \*tPtr +=3;  break;  case 3:  \*tPtr +=3;  break;  case 4:  \*tPtr +=3;  break;  case 5:  if(\*tPtr >= 6)  \*tPtr -=6;  else  \*tPtr =0;  break;  case 6:  if(\*tPtr >= 6)  \*tPtr -=6;  else  \*tPtr =0;  break;  case 7:  \*tPtr += 1;  break;  case 8:  \*tPtr += 1;  break;  case 9:  \*tPtr += 1;  break;  }  }  void moveHare(int \*hPtr)  {  switch (static\_cast<int>(rand())%10)  {  case 2:  \*hPtr += 9;  break;  case 3:  \*hPtr += 9;  break;  case 4:  if(\*hPtr >=12)  \*hPtr -= 12;  else  \*hPtr = 0;  break;  case 5:  \*hPtr -= 2;  break;  case 6:  \*hPtr -= 2;  break;  case 7:  \*hPtr += 1;  break;  case 8:  \*hPtr += 1;  break;  case 9:  \*hPtr += 1;  break;  }  }  bool moveLine(int \*tPtr,int \*hPtr,char \*lPtr)  {  if(\*tPtr != \*hPtr)  {  \*(lPtr + \*tPtr) = 'T';  \*(lPtr + \*hPtr) = 'H';  return 1;  }  else  return 0;  }  void clearLine(int \*tPtr,int \*hPtr,char \*lPtr)  {  \*(lPtr + \*tPtr) = ' ';  \*(lPtr + \*hPtr) = ' ';  }**四．界面显示**  BANG !!!!!  AND THEY ARE OFF !!!!!  H T  T  T  T H  T H  H T  H T  H T  H T  H T  H T  T H  T H  TH  H T  OUCH!!!  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T H  T  Hare wins. Yuch.  Process returned 0 (0x0) execution time : 0.156 s  Press any key to continue. | | | | | |
| |  | | --- | | **五．批复** | | | | | | |