Raymond Lin

Smallberg

CS31

27 November, 2017

Project 6 - Homework

* 1. int main()

{

int arr[3] = { 5, 10, 15 };

int\* ptr = arr;

\*ptr = 30; // set arr[0] to 30

\*(ptr + 1) = 20; // set arr[1] to 20

ptr += 2;

\*ptr = 10; // set arr[2] to 10

ptr = arr;

while (ptr <= arr + 2)

{

cout << \*ptr << endl; // print values

ptr++;

}

}

* 1. The program will not work because pToMax is a local variable. It must pass a reference instead.  
      void findMax(int arr[], int n, int\* &pToMax)  
      {  
      if (n <= 0)   
      return; // no items, no maximum!  
        
      pToMax = arr;  
       
      for (int i = 1; i < n; i++)  
      {  
      if (arr[i] > \*pToMax)  
      pToMax = arr + i;  
      }  
      }   
       
      int main()  
      {  
      int nums[4] = { 5, 3, 15, 6 };  
      int\* ptr;  
       
      findMax(nums, 4, ptr);  
      cout << "The maximum is at address " << ptr << endl;  
      cout << "It's at position " << ptr - nums << endl;  
      cout << "Its value is " << \*ptr << endl;  
      }
  2. The program does not work because ptr has not been initialized. To fix this, we must assign ptr to an address of a local variable.  
     void computeCube(int n, int\* ncubed)  
      {  
      \*ncubed = n \* n \* n;  
      }  
       
      int main()  
      {  
      int x = 5;  
      int\* ptr = &x;  
      computeCube(x, ptr);  
      cout << "Five cubed is " << \*ptr << endl;  
      }
  3. The program will not work because it compares two cstrings directly. The proper way to implement it would be to change them to the object that the pointer points to, using the star operator.  
      // return true if two C strings are equal  
      bool strequal(const char str1[], const char str2[])  
      {  
      while (\*str1 != ‘\0’ && \*str2 != ‘\0’)  
      {  
      if (\*str1 != \*str2) // compare corresponding characters  
      return false;  
      str1++; // advance to the next character  
      str2++;  
      }  
      return \*str1 == \*str2; // both ended at same time?  
      }  
       
      int main()  
      {  
      char a[15] = "Zhou";  
      char b[15] = "Zhu";  
       
      if (strequal(a,b))  
      cout << "They're the same person!\n";  
      }
  4. The program declares anArray within the function, and therefore it is a local variable.
  5. double\* cat;
  6. double mouse[5];
  7. cat += 4;
  8. \*cat = 25;
  9. \*(mouse + 3) = 42;
  10. cat -= 3;
  11. cat[1] = 54;
  12. cat[0] = 27;
  13. bool b = (\*cat == \*(cat + 1));
  14. bool d = (cat == mouse);
  15. double mean(const double\* scores, int numScores)  
       {  
       const double\* ptr = scores;  
       double tot = 0;  
       int i = 0;  
       while (i != numScores)  
       {  
       tot += \*(ptr + i);  
       i++;  
       }  
       return tot/numScores;  
       }
  16. const char\* findTheChar(char\* str, char chr)  
       {  
       for (int k = 0; \*(str + k) != ‘\0’; k++)  
       if (\*(str + k) == chr)  
       return str + k;  
        
       return nullptr;  
       }
  17. const char\* findTheChar(char\* str, char chr)  
       {  
       for (char\* c = str; \*c != ‘\0’ ; c++)  
       if (\*c == chr)  
       return c;  
        
       return nullptr;  
       }

1. 3  
   4  
   79  
   -1  
   9  
   22  
   19  
   Pointer ptr points to the first element of array, as a result of the maxwell function. Then it sets the first element to -1. Then it is incremented to point to the integer at position 2 of array. From there, position 2 of array is equal to position 0 of ptr. Position 3 of array is equal to position 1 of ptr, which is assigned the value 9. Then it prints a pointer to position 5 of array subtracted by pointer ptr. Because ptr still points to position 2 of the array, it prints the difference between the two positions, which is 3. Swap1 does not do anything because the pointers now point to each other’s objects but neither of the objects are changed. Swap2 exchanges the integers at position 0 and 2. At position 1 is now 4 and at position 2 is now -1. Then the array is simply printed out.
2. void removeS(char\* c) {

for (; \*c != '\0'; c++) {

if (\*c == 'S' || \*c == 's') {

for (char\* ptr = c; \*ptr != '\0'; ptr++) {

\*ptr = \*(ptr + 1);

}

c--;

}

}

}