**Huazhong University of Science and Technology**

**"Computer Fundamentals and Programming (C++)"**

**experimental report**

**Experiment name: The fifth computer experiment Experiment hours: 16 Name: Chen Haorui Student ID: U201612696 Class: Information Management (Medical) 1601 Experiment date: 12.7-12.9 Teacher's signature:**

**1. Purpose of the experiment**

* Master the similarities and differences between character arrays, character pointers, and numeric arrays.
* Master the definition, declaration and call of self-defined functions.
* Master the difference between passing by value and passing by address of function parameters.
* Understand the meaning of recursive algorithms.
* Clarify the meaning of the scope and lifetime of variables.

**2. Experimental content**

**1. reader program**

1 ) #include <iostream>

#include <cmath> \_

using namespace std;

int main()

{ int a=10, \*p=&a;

char ch = ' a ' , \*q=& ch ;

char str [ ] = " abcd " ,\* qq = str ;

cout << \*p<< endl ;

cout <<\*q<< endl ;

cout <<\* qq << endl ;

cout << p<< endl ;

cout <<q<< endl ;

cout << qq << endl ;

return 0;

}

**Run the above program and interpret each output.**

1. Output the number 10 pointed to by the pointer \*p; 2. Output the character a pointed to by the pointer \*q; 3. The first address of the string abcd is stored in qq, and the first character a of the string is output by the pointer \* qq ; 4. Use p to output the address assigned to variable a; 5. Use q to access the string starting from the character in the first address as a whole, but it is a single character, so output garbled characters when accessing the address after it; 6. Use qq to access Access to the whole string starting from the characters in the first address, and end when encountering ' \0 ' .

2 ) #include <iostream>

#include <cmath> \_

using namespace std;

int main()

{ int age ;

char name[11];

cin >> age;

cin. get ( ) ;

cin.getline ( name,11);

cout <<age<< endl ;

cout <<name<< endl ;

return 0; }

**Run the above program, whether the result is correct, if not, please correct it and explain the reason.**

Incorrect, entering a number after age cannot correctly distinguish whether it is part of age or a string. Modification measures: Insert cin.get () between the two inputs to manually isolate the contents of the two inputs.

3 ) #include <iostream>

#include <cstring> \_

using namespace std;

int main( )

{

char rstr [10 ] = " abcdefgh ", dstr [10];

char \* p = rstr , \*q = dstr ;

strcpy ( dstr,rstr );

cout << "output string rstr : "<< rstr << endl ;

cout << "output string dstr : "<< dstr << endl ;

cout << "output string p: "<<p<< endl ;

cout << "output string q: "<<q<< endl ;

return 0;

}

**Read the above program, run it, and analyze its results. If the above program is modified as follows and re-run, what is the result? Please explain why.**

int main( )

{

char \* rstr = " abcdefgh ", ~~\* dstr~~ dstr [9] ;

strcpy ( dstr,rstr );

cout << "output string rstr : "<< rstr << endl ;

cout << "output string dstr : "<< dstr << endl ;

return 0;

}

**The function of the above program is to copy the string rstr to the target string dstr , whether the result of running the program is correct, if not, please correct it and explain the reason.**

\* rstr and \* dstr are both character pointers, which can only be used for output operations, and cannot be reassigned, so the program reports an error. Change the assigned array to the form of a character array, and then perform strcpy copy operation on it.

1. **programming questions**

1 ) Write two functions to find the greatest common divisor and least common multiple of two integers respectively, call these two functions in the main function, calculate the greatest common divisor and least common multiple of two numbers and output the result.

**Require:**

1. The original code of the program. **(paste directly here)**

#include <iostream> \_

using namespace std;

//1. Remove and divide

int maxdiv ( int a , int b )

{

int c , d , temp ;

c = ( a >= b ) ? a : b ;

d = ( a < b ) ? a : b ;

while ( c % d )

{

c %= d ;

temp = c;

c = d;

d = temp;

}

return d;

}

//2. Rolling and subtracting

int maxdiv ( int a , int b )

{

int c , d , temp ;

c = ( a >= b ) ? a : b ;

d = ( a < b ) ? a : b ;

while ( c != d )

{

c -= d;

if ( c < d )

{

temp = c;

c = d;

d = temp;

}

}

return d;

}

int minmul ( int a , int b )

{

int i , c = ( a >= b ) ? a : b ;

for ( i = c ; ; i ++ )

{

if ( i % a == 0 && i % b == 0 )

break;

}

return i ;

}

int main()

{

int a , b ;

cout << "Please enter the value of a, b:";

cin >> a >> b ;

cout << "The least common multiple is:" << minmul ( a , b ) << endl ;

cout << "Greatest common divisor is:" << maxdiv ( a , b ) << endl ;

return 0;

}

1. Description of the algorithm idea to solve this problem

1. Separately call out two functions to operate the maximum and minimum values; 2. The greatest common divisor is divided by rolling, and the least common multiple is calculated by loop. When the remainder of the two numbers is zero, the condition is met and the return value is returned;

1. Records of the debugging process (including errors and modifications)

1. Wrongly assign c and d to a and b after the first remainder; 2. In the greatest common divisor, use ( c >= d ) ? c for c and d successively : d; ( c < d ) ? c : When d forgets that the value of c has changed after the operation, the value in d is the new value of c

1. Summarize the knowledge points or algorithms used in this question

1. Function nesting; 2. Rolling and dividing; 3. Loop

**2)** Write a program to randomly generate 50 random integers between 100-200, and find out the prime numbers among them. **The function of judging the prime number is realized by the function** , and the function is called in the main function to output all the prime numbers.

**Require:**

1. The original code of the program. **(paste directly here)**

# include <iostream>

# include <ctime>

# include <cmath>

# include <cstdlib>

using namespace std;

int count = 0;

void judge ( int n )

{

int i , flag = 0 ;

for ( i = 2 ; i <= sqrt ( n ) ; i ++ )

if ( n % i == 0 )

flag = 1;

if ( flag == 0 )

{

cout << n << ' ' ;

count ++ ;

if ( ! ( count % 5 ) )

{

cout << endl ;

count = 0;

}

}

}

int main()

{

int a [ 50 ] , i , count = 0 ;

cout << "Output 50 random numbers:" << endl ;

for ( i = 0 ; i < 50 ; i ++ )

{

srand ( ( unsigned ) time ( 0 ) + i ) ;

a [ i ] = 100 + ( int ) ( rand() % 100 ) ;

cout << a [ i ] << ' ' ;

count ++ ;

if ( ! ( count % 10 ) )

{

cout << endl ;

count = 0;

}

}

cout << "Output the prime numbers:" << endl ;

for ( i = 0 ; i < 50 ; i ++ )

judge ( a [ i ] ) ;

cout << endl ;

return 0;

}

1. Description of the algorithm idea to solve this problem

1. Determine the prime number program template; 2. The counter controls the line break; 3. If the local variable and the global variable have the same name, the global variable will be shielded in the function; 4. The random number seed generates a random variable

1. Records of the debugging process (including errors and modifications)

1. The position of the counter in the called function is wrong, causing the newline to fail; 2. The relationship between the condition for outputting a prime number and the flag variable is reversed;

1. Summarize the knowledge points or algorithms used in this question

1. Call function; 2. Judge prime number template; 3. Counter

**3)** Write a function to convert a hexadecimal integer to a decimal integer. For example, input "A2" converts to 162.

Tips: The cycle converts each character read into the corresponding decimal number, for example, the value of '1' is 1, the value of 'A' is 10, and the value of 'F' is 15, and the converted number is calculated and processed.

**The prototype of a function can be declared as:**

int funChange (char s[]); //The function of the function is to convert the string stored in the s array into a decimal value and return it. The parameter should be passed by the array name, that is, the formal parameter group and the actual parameter group share the same address space.

**Require:**

1. The original code of the program. **(paste directly here)**

# include <iostream>

using namespace std;

int funChange ( char s [ ] , int n )

{

int dec , sum = 0 , i ;

i = 0;

while ( s [ i ] != '\0' )

{

if ( s [ i ] >= '0' && s [ i ] <= '9' )

{

dec = s[ i ] - '0';

}

else

if ( s [ i ] >= 'A' && s [ i ] <= 'F' )

dec = s[ i ] - 'A' + 10;

if ( s [ i ] >= 'a' && s [ i ] <= 'f' )

dec = s[ i ] - 'a' + 10;

sum = sum \* 16 + dec ;

i ++;

}

cout << sum ;

return sum ;

}

int main()

{

int dec ;

char model [20] ;

cout << "Please enter a string, note that it can only contain numbers and letters in af (case-insensitive), and cannot contain spaces\n";

cin.getline ( model , 20 ) ;

dec = funChange ( model , 20 ) ;

cout << "The corresponding decimal number is" << dec << endl ;

return 0;

}

1. } Description of algorithm ideas to solve this problem

1. An independent function completes the conversion from a character array to a number array; 2. The while loop judges whether the string is terminated; 3. The mathematical operation of converting hexadecimal numbers to decimal numbers

1. Records of the debugging process (including errors and modifications?)

1. When the conversion program uses the for statement to loop, the loop continues after the string reaches the end, resulting in garbled output data

1. Summarize the knowledge points or algorithms used in this question

1. Convert characters to numbers;

**4)** Write a function to verify Goldbach's conjecture that any sufficiently large even number can be expressed as the sum of two prime numbers. It is required to define two functions. One function judges whether an integer is a prime number, and the other function verifies Goldbach's conjecture, which decomposes an even number into two prime numbers and returns the two prime numbers. Use the nested call of the function to complete (that is, call the function of judging the prime number in the function of verifying Goldbach's conjecture). For example, display information such as 8=3+5 is completed in the main function.

**Require:**

1. The original code of the program. **(paste directly here)**

# include <iostream>

# include < math.h >

using namespace std;

int judge ( int n )

{

int i , flag = 1 ;

for ( i = 2 ; i <= sqrt ( n ) ; i ++ )

if ( ! ( n % i ) )

{

flag = 0;

break;

}

return flag;

}

int gede ( int aim , int & i )

{

for ( i = 2 ; i <= aim / 2 ; i ++ )

if ( judge ( i ) && judge ( aim - i ) )

{

return 1 ;

break;

}

return 0 ;

}

int main()

{

int aim , i = 0 ;

cout << "Please enter an even number to be judged\n";

cin >> aim ;

if ( gede ( aim , i ) )

cout << aim << '=' << i << '+' << aim - i << endl ;

return 0;

}

1. Description of the algorithm idea to solve this problem

1. The loop structure judges the prime number; 2. Two prime numbers are the necessary and sufficient conditions for verifying Goldbach’s conjecture, so the condition in if should be true for the function processing values of the two numbers; 3. After the Goldbach’s conjecture is established, it should be At least one prime number is returned, and pass by reference is used at this time.

1. Records of the debugging process (including errors and modifications?)

No errors occurred.

1. Summarize the knowledge points or algorithms used in this question

1. Function nesting; 2. Reference passing; 3. Prime number template; 4. State variables

**5)** Write a function to find the subscripts of the maximum and minimum values in any one-dimensional array element. It is required to input the value of the array elements in the main function, and output the maximum and minimum values. (Maximum and minimum only need to consider one)

**Tip:** This question requires passing the elements of a one-dimensional array to the function for processing. For the transfer of a large amount of data, the best way is to use the array name as the actual parameter. In this case, the formal parameter can be a pointer or It is an array, and the data of the actual parameter array can be directly processed through the formal parameter. The problem in this question is to obtain the maximum and minimum values of the array elements. There are two values in the function that need to be returned to the main function, so the return statement cannot be used, and pointers or reference parameters must be used for return.

**The prototype of a function can be declared as:**

void funMaxMin ( int a[], int n, int \*max, int \*min) //The function of the function gets the maximum and minimum subscripts of the array elements.

**Require:**

1. The original code of the program. **(paste directly here)**

# include <iostream>

using namespace std;

void findmax ( double judge [] , int n , double &max , int & submax )

{

int i ;

max = judge[0];

for ( i = 0 ; i < n ; i ++ )

if ( judge [ i ] > max )

{

max = judge[ i ];

submax = i ;

}

}

void findmin ( double judge [] , int n , double &min , int & submin )

{

int i ;

min = judge[0];

for ( i = 0 ; i < n ; i ++ )

if ( judge [ i ] < min )

{

min = judge [ i ] ;

submin = i ;

}

}

int main()

{

int i , submax = 0 , submin = 0 ;

double origin [20] , max , min ;

cout << "Enter the original array:";

for ( i = 0 ; i < 20 ; i ++ )

{

cin >> origin [ i ] ;

}

findmax ( origin , 20 , max , submax ) ;

findmin ( origin , 20 , min , submin ) ;

cout << "The maximum value is: " << max << '\t' << "The subscript is: " << submax << endl ;

cout << "The minimum value is: " << min << '\t' << "The subscript is: " << submin << endl ;

return 0;

}

1. Description of the algorithm idea to solve this problem

1. Perform the maximum value and subscript operation on the entire array;

1. Records of the debugging process (including errors and modifications?)

1. Will not write the format passed by the array

1. Summarize the knowledge points or algorithms used in this question

1. Array transfer; 2. Reference transfer; 3. Find the most value template

**6)** Write a recursive function to count the number of digits of any positive integer, and input this integer in the main function and output the statistical result.

**Require:**

1. The original code of the program. **(paste directly here)**

# include <iostream>

using namespace std;

int stat ( int num , int count )

{

if ( num == 0 )

return count;

count ++ ;

stat ( ( int ) num / 10 , count ) ;

}

int main()

{

char j;

int num , count ;

do

{

count = 0;

cin >> num ;

cout << stat ( num , count ) << endl ;

cout << "Continue to enter? Yes - enter Y; No - enter any other key\n";

cin >> j ;

}

while ( j == 'y' || j == 'Y' ) ;

return 0 ;

}

1. Description of the algorithm idea to solve this problem

1. Call recursively until the target number is modified to 0; 2. The counter counts the number of times the last digit is removed

1. Records of the debugging process (including errors and modifications?)

1. The recursive termination condition in the recursive function is unclear and cannot be written; 2. The counter is wrongly placed after the recursive statement, resulting in failure; 3. The counter will not be reset to zero after the main function loops

1. Summarize the knowledge points or algorithms used in this question

1. Recursive call; 2. Counter

7) Write a program that requires the ability to sort any numerical array, and then use the half-search method to query the number to be searched. It is required that the sorting and searching functions are respectively implemented with custom functions. **Requirement: Whether the information found can only be output and displayed in the main function.**

**Require:**

1. The original code of the program. **(paste directly here)**

# include <iostream>

using namespace std;

void senum ( double a [] , int n , double target , int &flag , int &mid )

{

int high = 4 , low = 0 ;

mid = (high + low) / 2;

while ( a [mid] != target && low <= high )

{

if ( a [mid] < target )

low = mid + 1;

else

high = mid - 1;

mid = (high + low) / 2;

}

if ( a [mid] == target )

flag = 1;

}

void size ( double a [] , int n )

{

double temp;

int noswap , j, i ;

for ( j = 0 ; j < n ; j ++ )

{

noswap = 0;

for ( i = 4 ; i > j ; i -- )

if ( a [ i ] < a [ i-1] )

{

temp = a[ i ];

a [ i ] = a[i-1];

a [i-1] = temp;

noswap = 1;

}

if ( noswap == 0 )

break;

}

}

int main()

{

int i , flag = 0 , mid ;

double a [5] , target ;

cout << "Enter the original array:\n";

for ( i = 0 ; i < 5 ; i ++ )

cin >> a [ i ] ;

size ( a , 5 ) ;

cout << "Output sorted array:\n";

for ( i = 0 ; i < 5 ; i ++ )

cout << a [ i ] << ' ' ;

cout << endl << "Enter the number to be searched:\n";

cin >> target ;

senum ( a , 5 , target , flag , mid ) ;

if ( flag == 1 )

cout << "Found! The subscript of the target number is " << mid << endl ;

else

cout << "Number of targets found!\n ";

return 0 ;

}

1. Description of the algorithm idea to solve this problem

1. The state variable determines the output content; 2. Check whether the array is successfully sorted

1. Records of the debugging process (including errors and modifications?)

1. I didn't think clearly which variables to deal with in the half-find program; 2. The format of the array transfer is not clear

1. Summarize the knowledge points or algorithms used in this question

1. Bubble sorting template; 2. Divide in half search template; 3. Pass by reference;

**3. Suggestions for improving the content, methods and means of this experiment, as well as experimental experience**

**Experimental experience includes** :

1) Which knowledge points have been mastered except those below

2) Which knowledge points are difficult to pass arrays, multi-parameter mixed references, recursive calls, and the operating range of character pointers

3) Suggestions for lectures I want to add a few lessons to review key points or lecture examples

4) Practice more remedial suggestions for lack of knowledge