—

1. Single Choice (20points). Please choose the best answer for each question.

1) Which of the following statements about C++ is correct?
A) C++ is a subset of C.
B) C++ is developed based on Java.
C) Any C++ program can be replaced by a C program with equivalent functionality.
D) C++ is the best programming language.

2) Which of the following is a valid user-defined identifier?
A) _x B) 9X C) x# D) else

3) What is the value of (double)(3/2)?
A) 1.5 B) 1.0 C) 2.0 D) 3.0

4) Which of the following math expression is correct?
A) 4x+x B) (x-4)/2(x-5) C) ++(2.0/9+x) 常量不能赋值 D) x+=(x+1)

5) Suppose char x = '9', what is the printout of cout << x+1<<endl ?
A) 0 B) 58 +的结果是整数或浮点数 C) 10 D) A

6) Which of the following expression exactly equals to -- i ?
A) i-- B) i+=1 C) ++i D) i=i-1

7) Suppose *x* is a variable of i*nt*, which of the following is a meaningful boolean expression?
A) 1 < x < 100 B) x = 0 必为false C) x D) (x = 1) || (x != 1)必为true

8) Which of the following might cause an error?
A) char text[] = "Good"; B) char text[] = {'G', 'o', 'o', 'd'}没有数字0结尾
C) char text[5] = {'G', 'o', 'o', 'd'}; D) char text[] = {'G', 'o', 'o', 'd', 0};

9) What is the value of "(double)1/3*3 = = 1"?
A) true B) false C) 1 D) maybe true, maybe false浮点数运算不精确，且在不同的计算机中可能有不同的结果

10) Which of the following statements about variable is not correct?
A) A local variable will be automatically initialized to zero if no initial value is given.
B) The address of a variable is the address of the first byte of the variable's memory space.
C) The type of a variable determines the size of memory space used by the variable.
D) It is possible that two variables have the same name.

11) Which of the following is NOT an advantage of using functions?
A) To hide detailed implementation from the client/caller.

B) To ease code reusing.
C) To make programs easier to read.
D) To make programs run faster.

12) Which of the following statements about function is correct?
A) A function can be defined inside another function.
B) The implementation of a function may appear after the function is called.as long as it is declared before it is used
C) A value can only be passed from a callee to its caller using a return statement.
D) A default parameter must be the last parameter in any parameter list.

13) To overload a function, you must define functions with different:
A) headers B) names C) return types D) signatures

14) Which of the following about function matching is not correct?
A) A function cannot be matched if it has a different name.
B) A function cannot be matched if it has a different signature. Match after conversion is not anexact match
C) A function cannot be matched if it has no return value. Your answer can be B or C
D) A function cannot be matched if it has not enough number of arguments.

15) Suppose you declare variables as below, which of the following statements is true?

```
int i = 10;
int *pi = &i;
int &ri = i;
```

A) *pi* is 10 B) *&i* is 10
C) *pi* contains the value of *i* D) *i* contains the address of *i*

16) Which of the following statements about array is correct?
A) An array is exactly a pointer to a sequence of variables.也勉强算对
B) The computer will automatically check if each array index is out of bounds in run-time.
C) When an array is used as an argument, only the address of the array will be passed.
D) When an array is used as an argument, its length must be passed as another argument.

17) How to initialize an array of 10 double 0's ?
A) double array[] = {0}; B) double array[10] = {0};
C) double array[] = 0; D) double array[10] = 0;

18) Which of the following statement is not correct?
A) The scope of a local variable is within the function it is defined.
B) The scope of a static variable is within the function it is defined.
C) The lifetime of a local variable is equal to that of the program.
D) The lifetime of a static variable is equal to that of the program.

19) Which of the following statements about recursion is correct?

A) A recursive function must call itself in its function body.

B) An iterative function is easier to be designed than a recursive function with the same functionationality.

C) Recursion runs slower (than iteration), so it should not be used unless the problem cannot be solved by iteration.

D) Any recursive function can be replaced by an iterative function with the same functionality.

20) In which of the following case can a file be opened to write?

A) The file does not exist.

B) The file is being used by another program for input.

C) The file is being used by another program for output

D) The file is being used by another program for both input and output

2. Error correction (20points).

Please find out the errors, including compilation and run-time errors, in the following C++ codes and describe how to correct them.

1) A program to find out the value in an array that is the closest to the average. (5 errors)

```
1  #include <iostream>
2  using namespace std;
3
4  void fun() {
5      int tests;
6      cin >> tests;
7      for (int i = 0; i < tests; ++ i) {
8          int N = 10;
9          double data[N];
10         double avg = 0;
11         for (int j = 1; i < N; ++ j) {
12             cin >> data[j];
13             avg += data[j];
14         }
15     }
16     int index;
17     double min = avg - data[0];
18     if (min < 0) min = -min;
19     for (int i = 1; i <= N; ++ i) {
20         double diff = avg - data[i];
21         if (diff < 0) diff = -diff;
22         if (min > diff) {
23             min = diff;
24             index = i;
25         }
26     }
27     return data[index];
28 }
```

line 4: void -> int

line 8: const int N = 10;

line 11: j = 0;

line 16: index = 0;

line 19: i < N

2) A program that replaces a specific character in a string with another one. (10 errors)

```
1  void replace(const char array[length], char from, char to) {
2
3      for (int i = 0; i < length; ++ i) {
4          if (array[0] == from) {
5              array[i] = to;
6          }
7      }
8  }
9
10 int main() {
11     int length = 9;
12     int array[length] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
13     replace(array, 'a', 'b');
14 }
```

line 1: const char array[]

line 2: length should be a parameter

line 5: array[i] is a constant

line 12: length is not a constant

line 12: array index out of bound

line 13: first argument: cannot convert int * to char *

3. Output analysis (20points). Analyze and write down the output for the following programs.

1) A program that handles numbers

```
1  #include <iostream>
2  using namespace std;
3
4  int global  = 1;
5
6  void test(int x) {
7      int local = 1;
8      static static1 = 1;
9      cout << global << " " << local << endl;
10     cout << static1 << " " << x << endl;
11     global += 1;
12     ++ local;
13     static1 = static1 + 1;
14 }
15
16 int main() {
17     int local = 1;
18     while (local < 4) {
19         ++ local;
20         if (local % 2) continue;
21         test(local);
22     }
23 }
```

1 1

1 2

2 1

2 4

2) A program that handles strings

```
#include <iostream>
using namespace std;

void function1(char *, char *);

int function2(char * string1) {
    for (int i = 0; ; ++ i) {
        if (string1[i] == '\0') return i;
    }
}

bool compare(char * string1, char * string2, int n) {
    for (int i = 0; i < n; ++ i) {
        if (string1[i] != string2[i]) return false;
    }
    return true;
}

int main() {
    char string1[] = "abcabcabcabc";
    char string2[] = "bca";
    function1(string1, string2);
}

void function1(char * string1, char * string2) {
    int t1 = function2(string1);
    int t2 = function2(string2);
    int count = 0;
    int * location = new int[t1 - t2 + 1];
    for (int i = 0; i < t1 - t2; ++ i) {
        if (compare(string1 + i, string2, t2)) {
            location[count] = i;
            ++ count;
        }
    }
    for (int i = 0; i < count; ++ i) {
        cout << location[i] << endl;
        for (int j = 0; j < location[i]; ++ j)
            cout << string1[j];
        for (int j = location[i] + t2; j < t1; ++ j)
            cout << string1[j];
        cout << endl;
    }
}
```

1

abcabcabc

4

abcabcabc

7

abcabcabc

4. Fill-in-the-blank. 20points.

The following program splits an integer into factors, e.g. 12 = 2 * 2 * 3 .
Please fill in the blanks in the follows program to complete them.

```
#include <iostream>
using namespace std;
```

```
3
4  void getFactors(int number, bool isTheFirstFactor) {
5      if ( ____________ ) return;
6      int factor = 2;
7      while ( ____________ )
8          ++ factor;
9      if (isTheFirstFactor)
10         cout << number << " = " << factor;
11     else
12         cout << " * " << factor;
13      ____________ ;
14 }
15
16 void repeatTask(int array[], int size) {
17     for (int i = 0; i < size; ++ i) {
18         getFactors( ____________ );
19         cout << endl;
20     }
21 }
22
23 int main() {
24     int a[] = {12, 8, 2, 6, 7};
25     repeatTask( ____________ );
26 }
```

number == 1

number % factor != 0

getFactors(number / factor)

array[i], true

a, 5

5. Concept Explanation (10pints).

Please explain the following concept in details.

1) What is function matching and ambiguous function call?
Please give a C example of ambiguous function call.

Function matching is a task for the compiler to determine the function to be called in each function call express. It is determined according to the function name and the types of its parameters in the function call expression.

If there is no exact match for a function, the compiler will look for match after conversions of arguments. In the case of no exact match and there are multiple matches after conversion of argument, the compiler cannot determine which function to be called in the function call expression. This is when the compiler reports a *ambiguous function* call error.

```
1  int add(int a, int b) {
2      return a + b;
3  }
4
5  double add(double a, double b) {
6      return a + b;
7  }
8
9  int main() {
10     // call of overloaded 'add(int, double)'
11     // is ambiguous
12     int b = add(1, 2.0);
13 }
```

2) What are the relations (in terms of both commonalities and differences) between arrays and pointers (address variables)?

Please explain in terms of both their types and the operators they can use.

Please also show a C example for each thing that you explain.

The commonalities between array and pointer:

they are both address,

and the operator +, *, and [] are applicable to them.

The difference is that:

array is a constant address and pointer is a address variable,

and, therefore, the left operand of += and only be a pointer.

```
int array[] = {1, 2, 3};
int * pointer = array;
// + is applicable to both
cout << *(array + 1) << endl;
cout << *(pointer + 1) << endl;
// * is applicable to both
cout << *array << endl;
cout << *pointer << endl;
// [] is applicable to both
cout << array[1] << endl;
cout << *pointer[1] << endl;

// However, the left operand of
// += can only be a pointer
pointer += 1;
```

6. Programming Design (20points).

1. Implement the following function that removes all consecutive integers in a string

   void removeDigits(char string[])

   A main function shown in the following that calls this function will have the following output

```
#include <iostream>
using namespace std;

int main() {
    char string[] = "123adfad1234fasdf2134";
    removeDigits(string);
    cout << string;
}
```

OUTPUT

adfadfasdf

```
int length(char string[]) {
    for (int i = 0; ; ++ i) {
        if (string[i] == 0) return i;
    }
}

void remove(char string[], int index) {
    int len = length(string);
    for (int i = index; i < len; ++ i) {
        string[i] = string[i + 1];
    }
}

bool isDigit(char c) {
    return (c >= '0' && c <= '9');
}

void removeDigits(char string[]) {
    for (int i = length(string) - 1; i >= 0 ; -- i) {
        if (isDigit(string[i])) remove(string, i);
    }
}
```

2. Write a program to get all of the integer solutions that satisfy the following equations.

$$\begin{cases} x^2 - 20x + 75 = 0 \\ x \,\%\, 5 = 0 \\ 0 < x < 30 \end{cases}$$

OUTPUT

x = 5

x = 15

```
#include <iostream>
using namespace std;

bool isSolution1(int x) {
    return (x * x - 20 * x + 75 == 0);
}

bool isSolution2(int x) {
    return (x % 5 == 0);
}

int main() {
    for (int x = 1; x < 20; ++ x) {
        if (isSolution1(x) && isSolution2(x)) {
            cout << "x = " << x << endl;
        }
    }
}
```