Weekly Report(Jan.17,2018-Feb.4,2018)

张源娣

In the last three weeks,I have finished the courses about the C programming language on the MOOC,including the beginning and enhancing of it.

The followings are what I have learned.

Initially,Types,Operators,and Expressions.

0.Variable can be described as a place to save data.In C,all variables must be declared before they are used,usually at the beginning of the function before any executable statements.The declaration consists of a type name and a list of variables.There are several types.The types are int(means integer),float(means floating point), char(means character),short(means short integer),long(means long integer) and double(means double-precision floating point).Besides,there is another type named loop,which have two opposite values(True and False,or 1 and 0).We should include <stdbool.h> to use bool.

1.Apart from variables,there is another data type named constant quantity.We can declare a constant quantity by using ‘const’ at the very beginning.

2.There are some Arithmetic Operators,like ’%’,’/’,etc.But there isn’t symbol of exponentiation.To do this,we need to invoke the function named pow() instead of using ‘\*\*’,which is different from Python.And symbols like‘>’,’>=’,’==’are relational and logical operators.’++’,’--’are increment and decrement operators.Symbols like ’&’,’<<’,’~’are bitwise operators.

3.And one thing should be given priority to is that the type of the operating result of integers is still int. Thus,don’t forget type conversion when needed.

4.To exchange the values of two variables,we need to make it by dint of another variable.For example:

int a=5,b=6;

int t=a;

a=b;

b=t;

5.The range of both int and float depends on the machine we are using(I have no idea about it for I don’t have a deep knowledge about CPU);16-bit ints,which lie between -32768 and +32767,are common,as are 32-bit ints.A float number is typically a 32-bit quantity,with at least six significant digits and magnitude generally between about 10-38 and 10+38.And we can use sizeof() to find out how many bytes these types occupy.

Secondly,Control Flow.

0.if(expression)

{statement}

else if(expression)

{statement}

else if(expression)

{statement}

else

{statement}

We usually use ‘==’,’!=’,’>’,’>=’,’<’,’<=’in the expression.If the expression is True(1),the statement following will be operated,and the other statements will be omitted.

1.switch(expression)

{

case const-expr:statements;

case const-expr:statements;

default:statements;

}

Use break when needed.

2.while(expression)

{statement}

The expression is evaluated.If it is non-zero,statement is executed and expression is re-evaluated.The cycle continues until expression becomes zero,at which point execution resumes after statement.

3.for(expr1;expr2;expr3)

{statement}

It is equivalent to

expr1;

while(expr2)

{

statement;

expr3;

}

Sometimes,omitting the expr is acceptable.When the times of operation is known before,using for is helpful.

4.do

{statement}while(expression);

The expression is after the statement,which makes it possible to operate at least once.

5.Break and Continue

The break statement provides an early exit from for,while,and do,just as from switch.It causes the innermost enclosing loop or switch to be exited immediately.The continue statement is related to break,but less often used;it causes the next iteration of the enclosing for,while,or do loop to begin.A continue inside a switch inside a loop causes the next loop iteration.

6.Goto and Labels

C provides the infinitely-abusable goto statement,and labels to branch to.But we don’t use it quite often for thd single exit principle.

Thirdly, Functions

0.Functions make it more clear and easier for us when solving a complicated problems.To create a function,we use this form:

type name (some variables we will use)

{statement}

The type is the type of the variable the function will return.If there is no need for return or parameter,use void.And declaration is needed in the beginning if the function is after the main().

1.Local variables are declared in the blocks,which exist when the blocks operate.

And there is also a kind named external variables,which are defined outside of any function,and are thus potentially available to many functions.

2.The static declaration,applied to an external variable or function,limits the scope of that object to the rest of the source file being compiled.Static storage is specified by prefixing the normal declaration with the word static.It can also be applied to internal variables.Internal static variables are local to a particular function.

Forthly,Pointers and Arrays.

0.A pointer is a variable that contains the address of a varible.The unary operator & gives the address of an object,And the unary operator \* is the indirection or dereferencing operator;when applied to a pointer,it accessed the object the pointer points to.Pointer arguments enable a function to access and change objects in the function that called it.

1.There is a strong relationship between arrays and pointers.We can define an array like this:type name[numbers of elements].The index begins at 0.To make a[n] as an example,we can use sizeof(a)/sizeof(a[0]) to get the length.Sometimes we don’t need to write the number of element is we write all the elements in the right,and if the number of elements written is less than the actual number,the other elements will be initialized as 0.We can also define dyadic array by using type name[row][column].

2.We can include <stdlib.h> to apply for more places.Use void\* malloc(size\_t size);.But don’t forget free() the places applied later.

3.Character string exists as the form of array,which ends at 0(or ‘\0’).O is the symbol of ending,thus not included when counting the length of string.We can define a character string like:char \*str=’Hello’ or char word[]=’Hello’ or char line[10]=’Hello’.C provides some functions for string,like strlen() and strcmp(),to name just a few.

Fifthly,Structures.

0.A structure is a collection of one or more variables.It helps to organize complicated data,particularly in large programs.For example,two integers can be placed in a structure declared like this:

struct point{ struct { struct point{

int x; or int x; or int x;

int y; int y; int y;}

}p1,p2; }p1,p2; struct point p1,p2;

1.Unlike array,when using structure,if we want use pointers,the operator & is a must.Like this:struct date \*pDate=&today.

3.We can use typedef to declare a kind of existing type to simplify some names.

4.A union is a variable that may hold objects of different types and sizes,with the compiler keeping track of size and alignment requirements.The syntax is based on structures:

union u\_tag{

int ival;

float fval;

char \*sval;

}u;

Lastly,Input and Output.

0.The input and output functions,types,and macros defined in <stdio.h> .

1.FILE \*fopen(),FILE \*freopen(),fflush(),fseek() are some functions dealing with files.The printf functions provide formatted output conversation.And the scanf functions deal with formatted input conversion.

2.Using correct character is very important for formatted output and input.(I forgot using & when using scanf once,thus making a little problem for my computer,and it took me a long time to find out the reason.)

What I have finished these days is just basic,so here is my plan:

0.Try to solve some more difficult problems about the C programming language on the PTA website.

1.Read the book A Byte of Python to strengthen my studying of Python.

2.Learn the courses about Linear Algebra.