

C语言理论题错题集

2019年10月21日 18:47

A 1

7. Variable name resolving (number of significant characters for uniqueness of variable) depends on [1分]

- A Compiler and linker implementations
- B Assemblers and loaders implementations
- C C language
- D None

考生答案: C ×

参考答案: A

查看解析

我要纠错

这个resolving可以理解为解析，那当然是编译器做的嘛。最初理解为决定觉得和定义有点像

5. Variable names beginning with underscore is not encouraged. Why? [1分]

- A It is not standardized
- B To avoid conflicts since assemblers and loaders use such names
- C To avoid conflicts since library routines use such names
- D To avoid conflicts with environment variables of an operating system

考生答案: B ×

参考答案: C

查看解析

我要纠错

这个不知道，忘记汇编里面用不用下划线

A 2

7. Comment on the output of this C code?

```
#include <stdio.h>
int main()
{
char c;
int i = 0;
FILE *file;
file = fopen("test.txt", "w+");
fprintf(file, "%c", 'a');
fprintf(file, "%c", -1);
fprintf(file, "%c", 'b');
fclose(file);
file = fopen("test.txt", "r");
while ((c = fgetc(file)) != -1)    觉得-1用%c会变成正数
printf("%c", c);
return 0;
}[1分]
```

- A a
- B Infinite loop
- C Depends on what fgetc returns
- D Depends on the compiler

考生答案: C ×

参考答案: A

一开始以为char是unsigned的

A 3

*

2. What is the output of this C code?

```
#include <stdio.h>
int main()
{
    printf("C programming %s", "Class by\n%s Sanfoundry", "WOW");
}
```

[1分]

- A C programming Class by
WOW Sanfoundry
- B C programming Class by\n%s Sanfoundry
- C C programming Class by
%s Sanfoundry
- D Compilation error

我的答案: B

5. What is the output of this C code?

```
#include <stdio.h>
int main()
{
    int var = 010;           八进制
    printf("%d", var);      如果用0x? ? ? 是十六进制
}
```

[1分]

- A 2
- B 8
- C 9
- D 10

我的答案: B

3. For the following code snippet:

```
char *str = "Sanfoundry.com\0" "training classes";
The character pointer str holds reference to string:[1分]
```

- A Sanfoundry.com
- B Sanfoundry.com\0training classes
- C Sanfoundry.comtraining classes
- D Invalid declaration

考生答案: D X

参考答案: B

查看解析

我要纠错

字符串拼接

是个feature

A4

2. Comment on the output of this C code?

```
#include <stdio.h>
int main()
{
    const int i = 10;
    int *ptr = &i;
    *ptr = 20;
    printf("%d\n", i);
    return 0;
}[1分]
```

强制类型转换

- A Compile time error
- B Compile time warning and printf displays 20
- C Undefined behaviour
- D 10

考生答案: B ✓

查看解析

我要纠错

```
#include <stdio.h>
int main()
{
    char *var = "Advanced Training in C by Sanfoundry.com";
}
```

Which of the following format identifier can never be used for the variable var?[1分]

- A %f
- B %d
- C %c
- D %s

%d与%p的区别是d是十进制，p是十六进制

考生答案: A ✓

查看解析

我要纠错

2. What is the output of this C code?

```
#include <stdio.h>
int main()
{
    int i = 3;
    int l = i / -2;
    int k = i % -2;
    printf("%d %d\n", l, k);
    return 0;
}[1分]
```

- A Compile time error
- B -1 1
- C 1 -1
- D Implementation defined

考生答案: B ✓

3 The usual arithmetic conversions are performed on the operands.

4 The result of the binary * operator is the product of the operands.

5 The result of the / operator is the quotient from the division of the first operand by the second; the result of the % operator is the remainder. In both operations, if the value of the second operand is zero, the behavior is undefined.

6 When integers are divided, the result of the / operator is the algebraic quotient with any fractional part discarded.⁹³⁾ If the quotient a/b is representable, the expression $(a/b) * b + a \% b$ shall equal a .

93) This is often called “truncation toward zero”.

A6
A7

1. What is the output of this C code?

```
#include <stdio.h>
void main()
{
float x = 0.1;
if (x == 0.1)
printf("Sanfoundry");
else
printf("Advanced C Classes");
}[1分]
```

- A Advanced C Classes
- B Sanfoundry
- C Run time error
- D Compile time error

我的答案: A

```
#include <stdio.h>
int main()
{
    printf("%d", (float)0.1==(double)0.1);
    return 0;
}[1分]
```

浮点会全部转换为double再比较
float转换成double会有精度损失

6. When double is converted to float, the value is?[1分]

- A Truncated
- B Rounded
- C Depends on the compiler
- D Depends on the standard

考生答案: A X

参考答案: C

我要纠错

真不知道

8. What is the output of this C code?

```
#include <stdio.h>
int main()
{
int i = 23;
char c = -23;
if (i < c)           Char 不是unsigned的
printf("Yes\n");
else
printf("No\n");
}[1分]
```

- A Yes
- B No
- C Depends on the compiler
- D Depends on the standard

考生答案: A X

参考答案: B

我要纠错

A8

1. What is the difference between the following 2 codes?

```
#include <stdio.h> //Program 1
int main()
{
    int d, a = 1, b = 2;
    d = a++ + ++b;
    printf("%d %d %d", d, a, b);
}

#include <stdio.h> //Program 2
int main()
{
    int d, a = 1, b = 2;
    d = a++ + +b;
    printf("%d %d %d", d, a, b);
}
```

[1分]

- A No difference as space doesn't make any difference, values of a, b, d are same in both the case
- B Space does make a difference, values of a, b, d are different
- C Program 1 has syntax error, program 2 is not
- D Program 2 has syntax error, program 1 is not

我的答案: **D**

- 6 EXAMPLE 2 The program fragment **x+++++y** is parsed as **x + + + + + y**, which violates a constraint on increment operators, even though the parse **x + + + + y** might yield a correct expression.

8. Comment on the output of this C code?

```
#include <stdio.h>
int main()
{
    int i = 2;
    i = i++ + i;
    printf("%d\n", i);
}
```

[1分] **+ +是一定有side effect的**

- A = operator is not a sequence point
- B ++ operator may return value with or without side effects
- C it can be evaluated as (i++) + i or i + (++i)
- D Both a and b

考生答案: **D** **X**

参考答案: A

我要纠错

A | |

3. What is the output of this C code?

```
#include <stdio.h>
int main()
{
    int x = 1;
    short int i = 2;
    float f = 3;
    if (sizeof((x == 2) ? f : i) == sizeof(float))
        printf("float\n");
    else if (sizeof((x == 2) ? f : i) == sizeof(short int))
        printf("short int\n");
}[1分]
```

- A float 好像是会按照更长的那个来算的
 B short int
 C Undefined behaviour
 D Compile time error

考生答案: D X

参考答案: A

查看解析

我要纠错

A12

3. In expression $i = g() + f()$, first function called depends on [1分]

- A Compiler
 B Associativity of () operator
 C Precedence of () and + operator
 D Left to write of the expression

考生答案: C X

参考答案: A

我要纠错

6.5.15 Conditional operator

Syntax

1 *conditional-expression:*
 logical-OR-expression
 logical-OR-expression ? *expression* : *conditional-expression*

Semantics

- 4 The first operand is evaluated; there is a sequence point between its evaluation and the evaluation of the second or third operand (whichever is evaluated). **The second operand is evaluated only if the first compares unequal to 0; the third operand is evaluated only if the first compares equal to 0;** the result is the value of the second or third operand (whichever is evaluated), converted to the type described below.⁹⁸
- 5 If both the second and third operands have arithmetic type, the result type that would be determined by the usual arithmetic conversions, were they applied to those two operands, is the type of the result. If both the operands have structure or union type, the result has that type. If both operands have void type, the result has void type.
- 6 If both the second and third operands are pointers or one is a null pointer constant and the other is a pointer, the result type is a pointer to a type qualified with all the type qualifiers of the types pointed-to by both operands. Furthermore, if both operands are pointers to compatible types or to differently qualified versions of compatible types, the result type is a pointer to an appropriately qualified version of the composite type; if one operand is a null pointer constant, the result has the type of the other operand; otherwise, one operand is a pointer to **void** or a qualified version of **void**, in which case the result type is a pointer to an appropriately qualified version of **void**.

9. What is the output of this C code?

```
#include <stdio.h>
int main()
{
    int x = 2, y = 0, l;
    int z;
    z = y = 1, l = x && y;
    printf("%d\n", l);
    return 0;
}[1分]
```

- A 0
 B 1
 C Undefined behaviour due to order of evaluation can be different
 D Compilation error

考生答案: D X

参考答案: B

我要纠错

憨憨题目和1分部分不清

4. What is the value of i and j in the below code?

```
#include <stdio.h>
int x = 0;
int main()
{
    int i = (f() + g()) || g();
    int j = g() || (f() + g());
}
int f()
{
    if (x == 0)
        return x + 1;
    else
        return x - 1;
}
int g()
{
    return x++;
}[1分]
```

A i value is 1 and j value is 1
 B i value is 0 and j value is 0
 C i value is 1 and j value is undefined
 D i and j value are undefined

考生答案: A ×

参考答案: D

我要纠错

10. What is the output of this C code?

```
#include <stdio.h>
int main()
{
    int y = 2;
    int z = y + (y = 10);
    printf("%d\n", z);
}[1分]
```

A 12
 B 20
 C 4
 D Either 12 or 20

考生答案: A ×

参考答案: B

我要纠错

6.5.14 Logical OR operator

Syntax

1 *logical-OR-expression:*
 logical-AND-expression
 logical-OR-expression **| |** *logical-AND-expression*

Constraints

- 2 Each of the operands shall have scalar type.

Semantics

- 3 The **| |** operator shall yield 1 if either of its operands compare unequal to 0; otherwise, it yields 0. The result has type **int**.

- 4 Unlike the bitwise **|** operator, the **| |** operator guarantees left-to-right evaluation; if the second operand is evaluated, there is a sequence point between the evaluations of the first and second operands. If the first operand compares unequal to 0, the second operand is not evaluated.

应该是括号内内容先执行了

4. What is the value of i and j in the below code?

```
#include <stdio.h>
int x = 0;
int main()
{
    int i = (f() + g()) || g(); //序列点
    int j = g() || (f() + g());
```

这题f先执行还是g先执行并不会影响i的结果

```

}
int f()
{
    if (x == 0)
        return x + 1;
    else
        return x - 1;
}
int g()
{
    return x++;
}[1分]
```

- A i value is 1 and j value is 1
- B i value is 0 and j value is 0
- C i value is 1 and j value is undefined
- D i and j value are undefined

考生答案: C X

参考答案: D

我要纠错

5. What is the value of i and j in the below code?

```
#include <stdio.h>
int x = 0;
int main()
{
    int i = (f() + g()) || g(); //bitwise or
    int j = g() || (f() + g()); //bitwise or
```

这里无序列点无法确定执行顺序

```

}
int f()
{
    if (x == 0)
        return x + 1;
    else
        return x - 1;
}
int g()
{
    return x++;
}[1分]
```

- A i value is 1 and j value is 1
- B i value is 0 and j value is 0
- C i value is 1 and j value is undefined
- D i and j value are undefined

考生答案: D X

参考答案: C

我要纠错

6. What is the output of this C code?

```
#include <stdio.h>
```

A13

2. What is the output of this C code?

```
#include <stdio.h>
int main()
{
    int x = 1, y = 2;
    if (x && y == 1)
        printf("true\n");
    else
        printf("false\n");
} [1分]
```

&&是sequence point &&之前的值一定会比&&后的值先evaluate, 甚至超越运算符的优先级

- A true
- B false
- C Compile time error
- D Undefined behaviour

考生答案: A X

参考答案: B

我要纠错

4. What is the output of this C code?

```
#include <stdio.h>
int main()
{
    int x = 3, y = 2;
    int z = x /= y %= 2;
    printf("%d\n", z);
} [1分]
```

- A 1
- B Compile time error
- C Floating point exception
- D Segmentation fault

考生答案: D X

参考答案: C

我要纠错

F.4 Floating to integer conversion

If the floating value is infinite or NaN or if the integral part of the floating value exceeds the range of the integer type, then the “invalid” floating-point exception is raised and the resulting value is unspecified. Whether conversion of non-integer floating values whose integral part is within the range of the integer type raises the “inexact” floating-point exception is unspecified.³¹³⁾

```
#include <stdio.h>
int main(){
    printf("%f\n", -4.0/0.0);
```

结果是inf, 不会报错

如果转换成整形则会由于标准浮点数错误的runtime-error并且终止程序

6. What is the output of this C code?

```
#include <stdio.h>
int main()
{
    int x = 3; //, y = 2;
    const int *p = &x;
    *p++;
    printf("%d\n", *p);
} [1分]
```

由于++和*都是右结合的运算符, 而且优先级相同, 所以从右向左运算
等于*(p++)

运算符	解释	结合方式
() [] -> .	括号(函数), 数组、两种结构成员访问	由左向右
! ~ ++ -- + -	否定, 按位否定, 增量、减量, 正负号,	由右向左
* & (类型) sizeof	间接, 取地址, 类型转换, 求大小	
% / *	乘, 除, 取模	由左向右
+ -	加, 减	由左向右
<< >>	左移, 右移	由左向右
<= >= >	小于, 小于等于, 大于等于, 大于	由左向右
== !=	等于, 不等于	由左向右
&	按位与	由左向右
^	按位异或	由左向右
	按位或	由左向右
&&	逻辑与	由左向右
	逻辑或	由左向右
? :	条件	由右向左
= += -= *= /=	各种赋值	由右向左
&= ^= = <= >=		
,	逗号(顺序)	由左向右

8. What is the output of this C code?

8. What is the output of this C code?

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
int x = 2, y = 0;
```

```
int z = (x && y) | 1;
```

```
printf("%d\n", z);
```

```
}[1分]
```

A 0

B 1

C Compile time error

D 2

考生答案: B ×

参考答案: C

我要纠错

&= ^= ~=	<<= >>=	
/	逗号 (顺序)	由左向右

按位取反是单目运算符, 优先级也很高

先执行

A14

5. What is the output of this C code?

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
double b = 8;
```

```
b++;
```

```
printf("%lf", b);
```

```
}[1分]
```

A 9.000000

B 9

C 9.0

D Run time error

我的答案: A

f, F

A **double** argument representing a floating-point number is converted to decimal notation in the style $[-]ddd.ddd$, where the number of digits after the decimal-point character is equal to the precision specification. If the precision is missing, it is taken as 6; if the precision is zero and the # flag is not specified, no decimal-point character appears. If a decimal-point character appears, at least one digit appears before it. The value is rounded to the appropriate number of digits.

A15

A16

2. What is the output of this C code?

```
#include <stdio.h>
printf("%d", sizeof('a'));
```

'a'实际上对应的是ascii值，替换后，显然为int类型

- A 1
- B 2
- C 4
- D None of the mentioned

考生答案: **C** ✓

我要纠错

4. What is the output of this C code?

```
#include <stdio.h>
union temp
{
    char a;
    char b;
    int c;
}t;
int main()
{
    printf("%d", sizeof(t));
    return 0;
}
```

Union类型的大小是能包含最大的大小的整数倍

- A 1
- B 2
- C 4
- D 6

考生答案: **C** ✓

我要纠错

6. The sizeof(void) in a 32-bit C is_____. [1分]

- A 0
- B 1
- C 2
- D 4

考生答案: **B** ✓

我要纠错

6.5.3.4 The `sizeof` operator

Constraints

- 1 The `sizeof` operator shall not be applied to an expression that has function type or an incomplete type, to the parenthesized name of such a type, or to an expression that designates a bit-field member.

- 4 The value of the result is implementation-defined, and its type (an unsigned integer type) is `size_t`, defined in `<stddef.h>` (and other headers).

7.17 Common definitions `<stddef.h>`

- 1 The following types and macros are defined in the standard header `<stddef.h>`. Some are also defined in other headers, as noted in their respective subclauses.
- 2 The types are

`ptrdiff_t`

which is the signed integer type of the result of subtracting two pointers:

GCC 标注

5.20 Arithmetic on void- and Function-Pointers

In GNU C, addition and subtraction operations are supported on pointers to `void` and on pointers to functions. This is done by treating the size of a `void` or of a function as 1.

A consequence of this is that `sizeof` is also allowed on `void` and on function types, and returns 1.

The option `-Wpointer-arith` requests a warning if these extensions are used.

<https://gcc.gnu.org/onlinedocs/gcc-4.4.2/gcc/Pointer-Arith.html#Pointer-Arith>

7. What type of value does `sizeof` return? [1分]

- A char
- B short
- C unsigned int
- D long

考生答案: A ×

参考答案: C

我要纠错

之前不知道

B I

3. Comment on the following code below

```
#include <stdio.h>
void main()
{
    int x = 5;
    if (true);
    printf("hello");
}[1分]
```

- A It will display hello
- B It will throw an error
- C Nothing will be displayed
- D Compiler dependent

考生答案: **D** ×

参考答案: B

我要纠错

B2

5. What is the output of this C code?

```
#include <stdio.h>
int main()
{
    int a = 1, b = 1;
    switch (a)
    {
        case a*b:
            printf("yes ");
        case a-b:
            printf("no\n");
            break;
    }
}[1分]
```

- A yes
- B no
- C Compile time error
- D yes no

我的答案: **C**

6.8.4.2 The **switch** statement

Constraints

- 1 The controlling expression of a **switch** statement shall have integer type.
- 2 If a **switch** statement has an associated **case** or **default** label within the scope of an identifier with a variably modified type, the entire **switch** statement shall be within the scope of that identifier.¹³⁸⁾
- 3 The expression of each **case** label shall be an integer constant expression and no two of the **case** constant expressions in the same **switch** statement shall have the same value after conversion. There may be at most one **default** label in a **switch** statement. (Any enclosed **switch** statement may have a **default** label or **case** constant expressions with values that duplicate **case** constant expressions in the enclosing **switch** statement.)

6. What is the output of this C code?

```
#include <stdio.h>
int main()
{
    int x = 97;
    switch (x)
    {
        case 'a':
            printf("yes ");
            break;
        case 97:           重复case错误
            printf("no\n");
            break;          标准中查不到duplicate case value error
    }
}
```

[1分]

- A yes
- B yes no
- C Duplicate case value error
- D Character case value error

考生答案: C ✓

B3

2. The correct syntax for running two variable for loop simultaneously is. [1分]

- A for (i = 0; i < n; i++)
 for (j = 0; j < n; j += 5)
- B for (i = 0, j = 0;i < n, j < n; i++, j += 5)
- C for (i = 0; i < n;i++) {}
 for (j = 0; j < n;j += 5) {}
- D None of the mentioned

考生答案: C ✗

参考答案: B

我要纠错 没看懂题意

B4

6. What is the output of this C code?

```
#include <stdio.h>
void main()
{
    int i = 2;
    do
    {
        printf("Hi");
    } while (i < 2)
}
```

[1分]

- A Compile time error
- B Hi Hi
- C Hi
- D Varies

考生答案: A



我要纠错

C1

2. What is the output of this C code?

```
#include <stdio.h>
int main()
{
    void foo(), f();
    f();
}
void foo()
{
    printf("2 ");
}
void f()
{
    printf("1 ");
}
foo();
}
```

[1分]

- A Compile time error as foo is local to main
- B 1 2
- C 2 1
- D Compile time error due to declaration of functions inside main

考生答案: A



参考答案: B

3. What is the output of this C code?

```
#include <stdio.h>
int main()
{
void foo();
void f()
{
foo();
}
f();
}
void foo()
{
printf("2 ");
}
```

[1分]

- A 2 2
- B 2
- C Compile time error
- D Depends on the compiler

考生答案: **C** X

参考答案: D

C2

2. Which of the following function declaration is illegal? [1分]

- A double func();
int main() {}
double func() {}允许空语句
- B double func() {};
int main() {}
- C int main()
{
double func();
}
double func() //statements}
- D None of the mentioned

考生答案: **B** X

参考答案: D

5. The output of the code below is

```
#include <stdio.h>
void main()
{
int k = m();
printf("%d", k);
}
void m()
{
printf("hello");
}[1分]
```

?? 好像是返回最后一个值

- A hello 5
- B Error
- C Nothing
- D Junk value

考生答案: B



参考答案: A

C3

2. What is the output of this C code?

```
#include <stdio.h>
int x;
void main()
{
printf("%d", x);
}[1分]
```

全局变量会自动初始化为0

- A Junk value
- B Run time error
- C 0
- D Undefined

考生答案: A



参考答案: C

我要纠错

3. What is the output of this C code?

```
#include <stdio.h>
int x = 5;
void main()
{
    int x = 3;
    printf("%d", x);
    {
        x = 4;
    }
    printf("%d", x);
}
```

没有定义新变量，还是赋值原来的变量

- A Run time error
- B 3 3
- C 3 5
- D 3 4

考生答案: B X

参考答案: D

5. Functions in C are ALWAYS:[1分]

- A Internal
- B External
- C Both Internal and External
- D External and Internal are not valid terms for functions

考生答案: C X

参考答案: B

你都不告诉我internal和external是什么意思。。。

C 4

2. What is the output of this C code (without linking the source file in which ary1 is defined)?

```
#include <stdio.h>
int main()
{
    extern ary1[];
    printf("scope rules\n");
}
```

???

- A scope rules
- B Linking error due to undefined reference
- C Compile time error because size of array is not provided
- D Compile time error because datatype of array is not provided

考生答案: D X

参考答案: A

我要纠错

3. What is the output of this C code after linking with source file having definition of ary1?

3. What is the output of this C code after linking with source file having definition of ary1?

```
#include <stdio.h>
int main()
{
extern ary1[];
printf("%d\n", ary1[0]);
}[1分]
```

A Value of ary1[0]
 B Compile time error due to multiple definition
 C Compile time error because size of array is not provided
 D Compile time error because datatype of array is not provided

考生答案: D ✓

我要纠错

5. Comment on the output of this C code?

```
#include <stdio.h>
int main()
{
int i;
for (i = 0;i < 5; i++)
int a = i;
printf("%d", a);
}[1分]
```

语句块中要求语句至少是statement
 A a is out of scope when printf is called
 B Redeclaration of a in same scope throws error
 C Syntax error in declaration of a
 D No errors, program will show the output 5

考生答案: A ✗

参考答案: C

C5

4. What is the output of this C code?

```
#include <stdio.h>
void main()
{
    static double x;
    int x;
    printf("x is %d", x);
}
```

两个x重名

- [1分]
- A Nothing
 - B 0
 - C Compile time error
 - D Junkvalue

考生答案: **D** ✕

参考答案: C

我要纠错

6. Which of following is not accepted in C? [1分]

- A static a = 10; //static as
- B static int func (int); //parameter as static
- C static static int a; //a static variable prefixed with stat
- D All of the mentioned

和extren一样可以没有类型

考生答案: **A** ✕

参考答案: C

我要纠错

7. Which of the following cannot be static in C? [1分]

- A Variables
- B Functions
- C Structures
- D None of the mentioned

考生答案: **C** ✕

参考答案: D

我要纠错

27

What you see here is the "[Implicit Int Rule](#)" at work. Simply put the rule says:

"A **variable declared** without an explicit type name is assumed to be of type `int`."

Note that this rule [was revoked in the c99 Standard](#)^[Ref 1]. However, depending on your compiler and its settings, the first example might compile with a warning, or will fail to compile(*with strict compilation settings*)



C6

1. What is the output of this C code?

```
#include <stdio.h>
int main()
{
register int i = 10;
int *p = &i;
*p = 11;
printf("%d %d\n", i, *p);
}[1分]
```

C语言register变量不能取地址

- A Depends on whether i is actually stored in machine register
- B 10 10
- C 11 11
- D Compile time error

考生答案: C X

参考答案: D

5. What is the output of this C code?

```
#include <stdio.h>
int main()
{
register const int i = 10;
i = 11;
printf("%d\n", i);
}[1分]
```

不能同时使用

- A 10
- B Compile time error
- C Undefined behaviour
- D 11

考生答案: D X

参考答案: B

C7

6.2.2 Linkages of identifiers

- 1 An identifier declared in different scopes or in the same scope more than once can be made to refer to the same object or function by a process called *linkage*.²⁹⁾ There are three kinds of linkage: external, internal, and none.
- 2 In the set of translation units and libraries that constitutes an entire program, each declaration of a particular identifier with *external linkage* denotes the same object or function. Within one translation unit, each declaration of an identifier with *internal linkage* denotes the same object or function. Each declaration of an identifier with *no linkage* denotes a unique entity.
- 3 If the declaration of a file scope identifier for an object or a function contains the storage-class specifier **static**, the identifier has *internal linkage*.³⁰⁾

²⁹⁾ There is no linkage between different identifiers.

⁴⁾ For an identifier declared with the storage-class specifier **extern** in a scope in which a

- 4 For an identifier declared with the storage-class specifier `extern` in a scope in which a prior declaration of that identifier is visible,³¹⁾ if the prior declaration specifies internal or external linkage, the linkage of the identifier at the later declaration is the same as the linkage specified at the prior declaration. If no prior declaration is visible, or if the prior declaration specifies no linkage, then the identifier has external linkage.
- 5 If the declaration of an identifier for a function has no storage-class specifier, its linkage is determined exactly as if it were declared with the storage-class specifier `extern`. If the declaration of an identifier for an object has file scope and no storage-class specifier, its linkage is external.
- 6 The following identifiers have no linkage: an identifier declared to be anything other than an object or a function; an identifier declared to be a function parameter; a block scope identifier for an object declared without the storage-class specifier `extern`.
- 7 If, within a translation unit, the same identifier appears with both internal and external linkage, the behavior is undefined.

Forward references: declarations (6.7), expressions (6.5), external definitions (6.9), statements (6.8).

7. What is the output of this C code?

```
#include <stdio.h>
int main()
{
    auto i = 10;
    const auto int *p = &i;
    printf("%d\n", i);
}
```

[1分]

- A 10
- B Compile time error
- C Depends on the standard
- D Depends on the compiler

考生答案: B ×

参考答案: A

What you see here is the "[Implicit Int Rule](#)" at work. Simply put the rule says:

27

"A **variable declared** without an explicit type name is assumed to be of type `int`."

▼

Note that this rule was revoked in the c99 Standard[Ref 1]. However, depending on your compiler and its settings, the first example might compile with a warning, or will fail to compile(*with strict compilation settings*)

✓

1. The scope of an automatic variable is:[1分]

- A Within the block it appears
- B Within the blocks of the block it appears
- C Until the end of program
- D Both (a) and (b)

考生答案: B ×

参考答案: D

hape题目

C 8

Description

2. #include is called [1分]
- A Preprocessor directive
 - B Inclusion directive
 - C File inclusion directive
 - D None of the mentioned

考生答案: B 

参考答案: A

查看解析

我要纠错

165) Thus, preprocessing directives are commonly called "lines". These "lines" have no other syntactic significance, as all white space is equivalent except in certain situations during preprocessing (see the # character string literal creation operator in 6.10.3.2, for example).

§6.10

Language

161

ISO/IEC 9899:201x

Committee Draft — April 12, 2011

N1570

invocation of a function-like macro.

- 3 A text line shall not begin with a # preprocessing token. A non-directive shall not begin with any of the directive names appearing in the syntax.
- 4 When in a group that is skipped (6.10.1), the directive syntax is relaxed to allow any sequence of preprocessing tokens to occur between the directive name and the following new-line character.

Constraints

- 5 The only white-space characters that shall appear between preprocessing tokens within a preprocessing directive (from just after the introducing # preprocessing token through just before the terminating new-line character) are space and horizontal-tab (including spaces that have replaced comments or possibly other white-space characters in translation phase 3).

Semantics

- 6 The implementation can process and skip sections of source files conditionally, include other source files, and replace macros. These capabilities are called *preprocessing*, because conceptually they occur before translation of the resulting translation unit.

4. What is the output of this C code?

```
#include <stdio.h>
#define foo(m, n) m * n = 10
int main()
{
    printf("in main\n");
}[1分]
```

- A In main
- B Compilation error as lvalue is required for the expression m*n=10
- C Preprocessor error as lvalue is required for the expression m*n=10
- D None of the mentioned

考生答案: D 

参考答案: A



C9

2. Which directory the compiler first looks for the file when using #include ? [1分]

- A Current directory where program is saved
- B C:COMPILERINCLUDE
- C S:SOURCEHEADERS
- D Both (b) and (c) simultaneously

??

考生答案: **D** X

参考答案: B

4. How is search done in #include and #include "somelibrary.h" according to C standard?

[1分]

- A When former is used, current directory is searched and when latter is used, standard directory is searched
- B When former is used, standard directory is searched and when latter is used, current directory is searched
- C When former is used, search is done in implementation defined manner and when latter is used, current directory is searched
- D For both, search for 'somelibrary' is done in implementation-defined places

???

考生答案: **C** X

参考答案: D

我要纠错

5. How is search done in #include and #include "somelibrary.h" normally or conventionally?

[1分]

- A When former is used, current directory is searched and when latter is used, standard directory is searched
- B When former is used, predefined directory is searched and when latter is used, current directory is searched and then predefined directories are searched
- C When former is used, search is done in implementation defined manner and latter is used to search current directory
- D For both, search for somelibrary is done in implementation-defined manner

考生答案: **C** X

参考答案: B

6. Can function definition be present in header files? [1分]

- A Yes
- B No
- C Depends on the compiler
- D Depends on the standard

考生答案: C ✕

参考答案: A

我要纠错

8. What is the output of this C code?

```
#include <stdio.h>
#define foo(m, n) m ## n
void myfunc();
int main()
{
myfunc();
}
void myfunc()
{
printf("%d\n", foo(2, 3));
}[1分]
```

- A 23
- B 2 3
- C Compile time error
- D Undefined behaviour

考生答案: A ✓

m'm'm

1 #运算符: 字符串化

对于带参宏, #运算符可以将参数字符串化。

```
1 #include <stdio.h>
2
3 #define A(a, b) #a##b
4 #define B(a)    #a
5
6 int main(void)
7 {
8     A(hello, nihao);
9     B(Aha);
10
11     return 0;
12 }
```

预处理后的结果:

```
1 int main(void)
2 {
3     "hello""nihao";
4     "Aha";
5
6     return 0;
7 }
8
```

<https://blog.csdn.net/slowisfastlemon/article/details/102703414>

当有#或者##时候 参数不展开

C | D

2 ##运算符: 预处理器粘合剂

```
1 //##define A(a) a    //没问题, 宏正常替换
2
3 //##define A(a) qa
4 //【大小写字母 0123456789 " ' $ _ 】--> 参数和这些字符连接, 参数不替换
5 //##define A(a) q##a //可以正常替换
6
7 //##define A(a,b) a b //没问题
8
9 //##define A(a,b) a,b //没问题
10
11 //参数之间连接在一起, 预处理器不进行参数替换
12 #define A(a,b) a##b
13
14 int main(void)
15 {
16     A(x,x);
17
18     return 0;
19 }
20 }
```

3 宏不展开的特殊情况

当参数的前面有#或者##运算符时, 参数不展开。

```
1 #define R B
2 #define A hello
3 #define B(x) x
4
5 #define C(x) #x
6 #define D(x, y) x##y
7
8 int main(void)
9 {
10     B(A);
11     C(A);
12     D(A, B);
13
14     B(R);      // 宏只会替换一次, 并不会进行递归替
15
16     return 0;
17 }
```

预处理后的结果:

```
1 int main(void)
2 {
3     hello;
4     "A";
5     AB;
6
7     B;
8
9     return 0;
10
11 }
```

2. What is the output of this C code?

```
#include <stdio.h>
#define foo(m, n) " m ## n "
int main()
{
    printf("%s\n", foo(k, 1));
}
```

- [1分]
- A k 1
 - B k1
 - C Compile time error
 - D m ## n

考生答案: B X

参考答案: D

我要纠错

D
|

6. How many number of pointer (*) does C have against a pointer variable declaration? [1分]

- A 7
- B 127
- C 255
- D No limits.

考生答案: A X

参考答案: D

2. What is the output of this C code?

```
#include <stdio.h>
void main()
{
    int k = 5;
    int *p = &k;
    int **m = &p;
    printf("%d%d%d\n", k, *p, **p);
}
```

- [1分]
- A 5 5 5
 - B 5 5 junk value
 - C 5 junk junk
 - D Compile time error

考生答案: B X

参考答案: D

4. What is the output of this C code?

```
#include <stdio.h>
void main()
{
    int a[3] = {1, 2, 3};
    int *p = a;
    int *r = &p;
    printf("%d", (**r));
}
```

- [1分]
- A 1
 - B Compile time error
 - C Address of a
 - D Junk value

考生答案: A X

参考答案: B

7. What is the output of this C code?

```
#include <stdio.h>
int main()
{
    int i = 10;
    int *p = &i;
    foo(&p);
    printf("%d ", *p); //函数中局部变量不存在了，返回值就不可行
}
void foo(int **const p)
{
    int j = 11;
    *p = &j;
    printf("%d ", **p);
}
```

???

- A 11 11 11
- B 11 11 Undefined-value
- C Compile time error
- D Segmentation fault/code-crash

考生答案: **C** ✕

参考答案: B

D2

9. Which of the following are correct syntaxes to send an array as a parameter to function:

- [1分]
- A func(&array)
 - B func(array)
 - C func(*array)
 - D func(array[size])

考生答案: **BD** ✕

参考答案: AB

D3

6. What is the output of the code given below?

```
#include <stdio.h>
void foo( int[] );
int main()
{
int ary[4] = {1, 2, 3, 4};
foo(ary);
printf("%d ", ary[0]);
}
void foo(int p[4])
{
int i = 10;
p = &i;
printf("%d ", p[0]);
}
```

[1分]

- A 10 10
- B Compile time error
- C 10 1
- D Undefined behaviour

考生答案: B X

参考答案: C

04

指针之间只能减不能加

D5

3. What is the output of this C code?

```
#include <stdio.h>
int main()
{
char *str = "hello, world!!\n";
char strc[] = "good morning\n";
strcpy(strc, str);
printf("%s\n", strc);
return 0;
}
```

- [1分]
- A hello, world!!
 - B Compile time error
 - C Undefined behaviour
 - D Segmentation fault

考生答案: D X

参考答案: C

我要纠错

7.21.2.3 The `strcpy` function

Synopsis

```
1 #include <string.h>
    char *strcpy(char * restrict s1,
                 const char * restrict s2);
```

Description

2 The `strcpy` function copies the string pointed to by `s2` (including the terminating null character) into the array pointed to by `s1`. If copying takes place between objects that overlap, the behavior is undefined.

Returns

3 The `strcpy` function returns the value of `s1`.

4. What is the output of this C code?

```
#include <stdio.h>
int main()
{
char *str = "hello, world\n";
str[5] = '.';
printf("%s\n", str);
return 0;
}[1分]
 A hello, world
 B hello, world
 C Compile error
 D Segmentation fault
```

考生答案: A 

参考答案: D

我要纠错

```
int main()
{
    char *p="hello world";
    p[0]='H';
    printf("%s\n",p);
    return 0;
}
运行结果会出现断错误,原因在于,*p="hello world" 这句仅仅声明了一个指针变量,指向字符串"hello world",
而"hello world"这个字符串程序没有给它分配空间,编译器把它分配到常量区.而常量字符串的值是不允许被修改的
,所以会出现断错误.
```

D 6

2. Which of the following declaration throw run-time error? [1分]

A int **c = &c;
 B int **c = &*c;
 C int **c = **c;
 D None of the mentioned

考生答案: C 

参考答案: D

我要纠错

) .

7. What is the output of this C code?

```
#include <stdio.h>
void main()
{
    int a[3] = {1, 2, 3};
    int *p = a;
    int **r = &p;
    printf("%d", (**r));
}
```

[1分]

- A 1
- B Compile time error
- C Address of a
- D Junk value

考生答案: A X

参考答案: B

我要纠错

→ int * r

D P

7. Comment on the 2 arrays regarding P and Q:

```
int *a1[8];
int *(a3[8]);
P. Array of pointers
Q. Pointer to an array [1分]
```

A2在哪里呢? ??

- A a1 is P, a2 is Q
- B a1 is P, a2 is P
- C a1 is Q, a2 is P
- D a1 is Q, a2 is Q

考生答案: A X

参考答案: B

D Q

1. To declare a 3 dimension array using pointers, which of the following is the correct

syntax: [1分]

- A char *a[][];
- B char **a[];
- C char ***a;
- D All of the mentioned

考生答案: D X

参考答案: A

>

.

2. Comment on the output of this C code?

```
#include <stdio.h>
int main()
{
    char *a = {"p", "r", "o", "g", "r", "a", "m"};
    printf("%s", a);
}[1分]
```

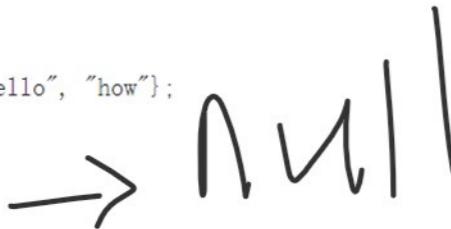
- A Output will be program
- B Output will be p
- C No output
- D Compile-time error

考生答案: A X

参考答案: B

4. What is the output of this C code?

```
#include <stdio.h>
void main()
{
    char *a[10] = {"hi", "hello", "how"};
    int i = 0;
    for (i = 0; i < 10; i++)
        printf("%s", *(a[i]));
}[1分]
```



- A Segmentation fault
- B hi hello how followed by 7 null values
- C 10 null values
- D depends on compiler

考生答案: B X

参考答案: A

手写答案

6. What is the output of this C code?

```
#include <stdio.h>
void main()
{
    char *a[10] = {"hi", "hello", "how"};
    printf("%d\n", sizeof(a));
}[1分]
```

A handwritten answer '4 x 10' written next to the question number 6.

- A 10
- B 13
- C Run time error
- D 40

考生答案: A X

参考答案: D

D 9

3. What is the output of this C code?

```
#include <stdio.h>
void main()
{
    char a[10][5] = {"hi", "hello", "fellows"};
    printf("%s", a[2]);
}
```

- A fellows
- B fellow
- C fello
- D fell

考生答案: A ×

参考答案: C

Answer: C

Explanation: Since every row in the array a[10][5] can contain only 5 characters, the a[2] element will hold "fello" i.e. 5 characters. There will not be any null character in a[2]. Since, the array is completely initialized, other rows (row a[3] will have only null characters. Hence, printf() using %s specifier will display fello only.

D 10

4. What type of array is generally generated in Command-line argument? [1分]

- A Single dimension array
- B 2-Dimensional Square Array
- C Jagged Array
- D 2-Dimensional Rectangular Array

考生答案: A ×

参考答案: C

我要纠错

交錯型數列
char * argv[]

D 1 |

2. How to call a function without using the function name to send parameters? [1分]

- A typedefs
- B Function pointer
- C Both (a) and (b)
- D None of the mentioned

考生答案: **C** 

参考答案: B

我要纠错

4. Which of the following is not possible in C? [1分]

- A Array of function pointer
- B Returning a function pointer
- C Comparison of function pointer
- D None of the mentioned

考生答案: **B** 

参考答案: D

5. What is the output of this C code?

```
#include <stdio.h>
void first()
{
    printf("Hello World");
}
void main()
{
    void *ptr() = first;
    ptr++;
    ptr();
}
```

正确方式: `void (*ptr)()=first;`

- A Illegal application of ++ to void data type
- B pointer function initialized like a variable
- C Both (a) and (b)
- D None of the mentioned

考生答案: **A** 

参考答案: C

8. What is the output of this C code?

```
#include <stdio.h>
void f(int (*x)(int));
int myfoo(int);
int (*fooptr)(int);
int ((*foo(int)))(int);
int main()
{
    fooptr = foo(0);
    fooptr(10);
}
int ((*foo(int i)))(int)
{
    return myfoo;
}
int myfoo(int i)
{
    printf("%d\n", i + 1);
}
```

[1分]

- A 10
- B 11
- C Compile time error
- D Undefined behaviour

考生答案: **C** X

参考答案: B

Int ((*foo(int i)))(int) == int (*foo(int i))(int)

E |

4. Which of the following cannot be a structure member? [1分]

- A Another structure
- B Function
- C Array
- D None of the mentioned

考生答案: **D** X

参考答案: B

能有function pointer 而不是function

2. User-defined data type can be derived by _____. [1分]

你说是就是, 换个名字就算新类型? ? ? ?

- A struct
- B enum
- C typedef
- D All of the mentioned

考生答案: **A** X

参考答案: D

5. Which of the following structure declaration will throw an error? [1分]
- A struct temp{} s;
 - B struct temp{};
struct temp s;
main()
 - C struct temp s;
struct temp{};
main()
 - D None of the mentioned
- 考生答案: C X
- 参考答案: C

E 2

8. The output of the code below is

```
#include <stdio.h>
struct student
{
    char *name;
};
struct student fun(void)
{
    struct student s;
    s.name = "alan";
    return s;
}
```

```
void main()
{
    struct student m = fun();
    s.name = "turing";
    printf("%s", m.name);
}
```

变量没声明。。。。

- [1分]
- A alan
 - B Turing
 - C Compile time error
 - D Nothing

考生答案: A X

参考答案: C

E 4

1. What is the output of this C code?

```
#include <stdio.h>
struct p
{
    int x;
    char y;
};
int main()
{
    struct p p1[] = {1, 92, 3, 94, 5, 96};
    struct p *ptr1 = p1;
    int x = (sizeof(p1) / 3);
    if (x == sizeof(int) + sizeof(char))
        printf("%d\n", ptr1->x);
    else
        printf("falsen");
}
```

= 24

E 6

7. What is the output of this C code?

```
#include <stdio.h>
struct p
{
char *name;
struct p *next;
};
struct p *ptrary[10];
int main()
{
struct p p, q;
p.name = "xyz";
p.next = NULL;
ptrary[0] = &p;
strcpy(q.name, p.name);
ptrary[1] = &q;
printf("%s\n", ptrary[1]->name);
return 0;
}[1分]
```

q还没申请空间

- A Compile time error
- B Segmentation fault/code crash
- C Depends on the compiler
- D xyz