试卷：2017秋-高级语言程序设计-期末试卷

**一、单项选择题(20分)**

1.(2分) 以下错误的描述是\_\_\_\_\_C\_\_\_\_\_\_\_\_

1. 使用while和do~while循环时，循环变量初始化的操作应在循环语句之前完成
2. while循环是先判断表达式，后执行循环体语句
3. do~while和for循环均是先执行循环体语句，后判断表达式
4. for、while、do~while循环中的循环体均可由空语句构成

2. (2分) 若有说明语句：int a,b,c,\*d=&c；，则能正确从键盘读入三个整数分别赋给变量a、b、c的语句是\_\_A\_\_\_\_\_

1. scanf("%d%d%d",&a,&b,d);
2. scanf("%d%d%d",&a,&b,&d);
3. scanf("%d%d%d",a,b,d);
4. scanf("%d%d%d",a,b,\*d);

3.(2分) 以下程序的输出结果是\_\_A\_\_\_\_\_

f(int b[],int m,int n)

{

int i,s=0;

for(i=m;i<n;i=i+2) s=s+b[i];

return s;

}

main()

{

int x,a[]={1,2,3,4,5,6,7,8,9};

x=f(a,3,7);

printf("%d\n",x);

}

1. 10
2. 18
3. 8
4. 15

4.(2分) 有以下程序

int fl(int x,int y)

{ return x>y?x:y; }

int f2(int x,int y)

{ return x>y?y:x; }

main()

{

int a=4,b=3,c=5,d,e,f；

d=f1(a,b); d=f1(d,c);

e=f2(a,b); e=f2(e,c);

f=a+b+c-d-e;

printf("%d,%d,%d\n",d,f,e);

}

执行后输出结果是\_\_\_\_C\_\_\_\_

1. 3,4,5
2. 5,3,4
3. 5,4,3
4. 3,5,4

5.(2分) 有以下程序

main()

{ char str[]="xyz",\*ps=str;

while(\*ps)ps++;

for(ps--;ps-str>=0;ps--) puts(ps);

}

执行后的输出结果是\_\_\_B\_\_\_\_\_

A) yz B) z C) z D) x

xyz yz yz xy

xyz xyz

A

B

C

D

6.(2分) 有以下程序段

int a[10]={1,2,3,4,5,6,7,8,9,10},\*p=&a[3],b;

b=p[5];

b中的值是\_\_\_D\_\_\_\_\_

1. 5
2. 6
3. 8
4. 9

7.(2分) 以下叙述中正确的是\_\_\_A\_\_\_\_\_

局部变量说明为static存储类，其生存期将得到延长

全局变量说明为static存储类，其作用域将被扩大

任何存储类的变量在未赋初值时，其值都是不确定的

函数调用可以作为一个函数的形参>>>>实参

8.(2分) 设已定义i和k为int类型变量，则以下for循环语句

for(i=0,k=-1;k=1; i++,k++)

printf( "\* \* \* \*\n");\_\_\_C\_\_\_\_\_

1. 判断循环结束的条件不合法
2. 是无限循环
3. 循环一次也不执行
4. 循环只执行一次

9.(2分) 已知字母A的ASCII码是65，以下程序的执行结果是\_\_\_\_D\_\_\_\_

#include<stdio.h>

main()

{ char c1='A', c2='Y';

printf(" %d,%d"\n",c1,c2);

}

1. A,Y
2. 65,65
3. 65,90
4. 65,89

10.(2分) 下列说法错误的是（A）。

1. C语言中的二维数组在内存中是按列存储的。
2. 在C语言中，数组的下标都是从0开始的。
3. 在C语言中，不带下标的数组名代表数组的首地址，即第一个元素在内存中的地址。
4. 不能使用变量定义数组的大小，但是在访问数组元素时在下标中可以使用变量或表达式。

**二、判断题（回答是，或者否）(10分)**

1.(1分) 函数的定义可以嵌套，但函数的调用不可以嵌套。\_\_\_\_F\_\_\_\_

2.(1分) 在函数调用过程中，不能通过改变形参的值而使实参的值也改变，但可以通过改变指针形参的值而使指针实参的值也改变。\_\_F\_\_\_\_\_\_

3.(1分) break和 continue语句都可用于选择结构和循环结构中。\_\_F\_\_\_\_\_\_

4.(1分) 若用数组名作为函数的实参，传递给形参的是数组第一个元素的值。\_\_\_F\_\_\_\_\_

5.(1分) 凡是函数中未指定存储类别的局部变量其隐含的存储类别是自动(auto)变量。T

6.(1分) C语言中规定函数的返回值的类型是由return语句中的表达式类型所决定。F

7.(1分) 在C语言中对文件操作必须先关闭当前文件，然后再打开一个新文件。\_\_\_T\_\_\_\_\_

8.(1分) 如果在程序中定义静态变量和全局变量时，未明确指明其初始值，那么它们可以在程序编译阶段自动被初始化为0值。\_\_\_\_T\_\_\_\_

9.(1分) 不同的函数中可以使用相同的变量名。\_\_\_T\_\_\_\_\_

10.(1分) C语言认为名为Student和student的变量是不同的变量。\_\_\_T\_\_\_\_\_

**三、一般填空题(20分)**

1.(5分) 写出下列程序的运行结果：

#include<stdio.h>

void Fun(int \*y)

{

printf("\*y=%d\n", \*y);

\*y = 20;

printf("\*y=%d\n", \*y);

}

main()

{

int x = 10;

printf("x=%d\n", x);

Fun(&x);

printf("x=%d\n", x);

}

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.(5分) 以下程序运行后的输出结果是

int a=5;

fun(int b)

{

static int a=10;

a+=b++;

printf("%d ",a);

}

main()

{

int c=20;

fun(c);

a+=c++;

printf("%d\n",a);

}

1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.(5分) 以下程序运行后的输出结果是 \_\_\_\_\_\_ 。

main()

{ int p[7]={11,13,14,15,16,17,18};

int i=0,j=0;

while(i<7 && p[i]%2==1)

j+=p[i++];

printf("%d\n",j);

}

4.(5分) 写出下面程序的输出结果

#include<stdio.h>

**struct** w

{

**char** low;

**char** high;

};

**union** u

{

**struct** w byte;

**short** word;

}uw;

**int** main( )

{

**int** result;

uw.word=0x1234;

**printf**("word value:%04x\n",uw.word);

**printf**("high byte:%02x\n",uw.byte.high);

**printf**("low byte:%02x\n",uw.byte.low);

uw.byte.low=0x74;

**printf**("word value:%04x\n",uw.word);

result=uw.word+0x2a34;

**printf**("the result:%04x\n",result);

**return** 0;

}

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**四、程序填空题(20分)**

1.(5分) 从键盘输入一个长整数，如果这个数是负数，则取它的绝对值，并显示出来。

#include <stdio.h>

main()

{

long int n;

printf("Enter the data:\n");

scanf(\_\_\_\_\_\_\_\_\_\_\_\_\_);

printf("\*\*\* the absolute value \*\*\*\n");

if (n < 0)

\_\_\_\_\_\_\_\_\_\_\_\_\_

printf("\n\n");

printf(\_\_\_\_\_\_\_\_\_\_\_\_\_);

}

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.(5分) 该程序的功能是：判断某一个年份是否为闰年。

#include <stdio.h>

int fun(int n)

{

int flag = 0;

if (n % 4 == 0)

{

if (\_\_\_\_\_\_\_\_\_\_\_\_\_)

flag = 1;

}

if (\_\_\_\_\_\_\_\_\_\_\_\_\_)

flag = 1;

return \_\_\_\_\_\_\_\_\_\_\_\_\_;

}

main()

{

int year;

printf("Input the year:");

scanf("%d", &year);

if (fun(year))

printf("%d is a leap year. \n", year);

else

printf("%d is not a leap year.\n",

year);

}

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.(5分) 下面程序模拟了骰子的6 000次投掷，用函数rand()产生1～6之间的随机数face，然后统计每一面1～6出现的机会（概率）存放

到数组frequency中。

#include <stdlib.h>

#include <time.h>

#include <stdio.h>

int main(void)

{

int face, roll, frequency[7] = {0};

srand(time (NULL));

for (roll = 1; roll <= 6000; roll++)

{

face = \_\_\_\_\_\_\_\_;

++\_\_\_\_\_\_\_\_;

}

printf("%4s%17s\n", "Face", "Frequency");

for (face = 1; face <= 6; face++)

{

printf("%4d%17d\n", face, frequency[face]);

}

return 0;

}

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4.(5分) 下面程序输入一行字符，然后统计其中有多少单词。假设单词之间以空格分开。

#include <stdio.h>

int CountWords(char str[]);

void main()

{

char str[20];

printf("Input a string:");

\_\_\_\_\_\_\_\_\_\_\_;

printf("Numbers of words = %d\n", CountWords(str));

}

int CountWords(char str[])

{

int i, num;

num = (str[0] != ' ') ? 1 : 0;

for (i=1;\_\_\_\_\_\_\_\_\_\_\_ ; i++)

{

if (\_\_\_\_\_\_\_\_\_\_\_)

{

num++;

}

}

return num;

}

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**五、普通编程题(30分)**

1.(8分) 从键盘输入某班学生某门课的成绩（每班人数最多不超过40人），当输入为负值时，表示输入结束，试编程计算并打印最高分。

\*\*输入格式要求："%d" 提示信息："Total students are %d\n" "The highest score is %d\n" "Input score:"

\*\*输出格式要求："%d"

程序的运行示例如下：

Input score:80

Input score:100

Input score:90

Input score:60

Input score:-1

Total students is 4

The highest score is 100

2.(10分) 分别按如下函数原形编程计算并输出求m\*m阶矩阵的转置矩阵。期中m的值由用户从键盘输入。已知m的值都不超过10。

输入函数原形：void InputMatrix(int a[][N], int m);

求转置矩阵函数原形：void Transpose(int a[][N], int at[][M], int m);

输出函数原形：void PrintMatrix(int at[][M], int m);

输入输出要求:

输入提示信息和格式信息：

"Input m:"

"%d "

"Input %d\*%d matrix:\n"

"%d"

输出提示信息和格式信息：

"The transposed matrix is:\n"

"%8d"

程序运行示例如下：

Input m:5

Input 5\*5 matrix:

1 2 3 4 5

6 7 8 9 10

11 12 13 14 15

16 17 18 19 20

21 22 23 24 25

The transposed matrix is:

1 6 11 16 21

2 7 12 17 22

3 8 13 18 23

4 9 14 19 24

5 10 15 20 25

3.(12分) 背景：钟面上的时针和分针之间的夹角总是在 0 度~ 359 度之间。举例来说，在十二点的时候两针之间的夹角为 0 度，而在六点的

时候夹角为 180 度，在三点的时候为 90 度。本题要解决的是计算 0:00 到 12:00之间任意一个时间的夹角。

输入：要求输入必须为 小时h分钟m的格式，如12h00m, 5h43m等，其他形式的输入都不接收，如 12,00 或5h43等输入都不被接收，而且需要

对“小时”和“分钟”进行数值有效判断（小时在[0,12],分钟在[0.60]之间），不满足以上要求时，需要重新输入。

在程序中首先打印：Please input time(e.g: 5h43m)，然后输入时间。

输出：对应每组测试数据，用常用格式显示时间以及这个时候时针和分针间的最小夹角，精确到小数点后一位。如：

输入“12h00m”时，输出 At 12:00 the angle is 0.0 degrees.

输入“5h43m” 时，输出 At 5:43 the angle is 86.5 degrees.

输入“5h5m” 时，输出 At 5:05 the angle is 122.5 degrees.

提示1：以表中心到12点的连线为基准，分针每走1分钟是6度，时针与基准的夹角每个小时也是30度，从整点开始，每过1分钟时针再增加0.5

度。要求结果角度为正值，即最终要取绝对值（fabs函数）

提示2：二者之间角度不应大于180度，如果大于，应用360度减去该角度。