

浙江大学 2013 - 2014 学年冬季学期

《程序设计基础》课程期末考试试卷

课程号: 211Z0040, 开课学院: 计算机学院

考试试卷: A 卷、B 卷 (请在选定项上打)

考试形式: 闭、开卷 (请在选定项上打)，允许带 / 入场

考试日期: 2014 年 01 月 12 日, 考试时间: 120 分钟

诚信考试, 沉着应考, 杜绝违纪.

考生姓名: _____ 学号: _____ 所属院系: _____

(注意: 答题内容必须写在答题卷上, 写在本试题卷上无效)

Section 1: Single Choice(2 marks for each item, total 20 marks)

- Which one below is NOT a keyword of the C programming language? _____
A. define B. if C. typedef D. while
- For the following declaration in a function:
 int i, j=6;
Which statement of the following is true? _____
A. both i and j has initial value of 6.
B. i will be initialized as zero, while j will be 6.
C. i will not be initialized, but j will get 6.
D. It will be error in compiling.
- For code below:
 unsigned short sht = 0;
 sht --;
What will be the value of **sht** after the execution? _____
A. -1 B. 65535 C. 32767 D. 65536
- Which literal below is 124 in decimal(十进制)? _____
A. 0124 B. 124d C. 0x124 D. 0174
- Given function prototype below:
 int* f(int *p);
and variable definition as:
 int i;
Which calling to this function is NOT correct? _____
A. f(&i); B. *f(&i) = 6; C. int* p = f(&i); D. f(&(i+6));
- Given an array declaration: **static int a[3][4]= {0}**; the description _____ in the following is correct.
A. Each element of the array **a** can have been initialized by 0.
B. The declaration is NOT correct.

- C. Each element of the array **a** can be initialized , but their initial values are NOT ALWAYS(不一定) 0.
- D. Only element **a[0][0]** can be initialized by 0.
7. The following code fragment will print out _____.
- ```
char s[]="student";
printf("%s%d", s+3, sizeof(s));
```
- A. student7      B. dent7      C. dent8      D. student8
8. The following code fragment will print out \_\_\_\_\_.
- ```
int *p, *q, k = 10, j=1 ;
p=&j; q = &k ; p = q ; (*p)++;
printf("%d",*q);
```
- A. 10 B. 11 C. 1 D. 2
9. Given the declaration: **int a[10], *p=a;**, the expression _____ is NOT correct.
- A. a[9] B. p[5] C. *p++ D. a++
10. In the following code fragments, item _____ is correct.
- A. int *p; scanf("%d", &p); B. int *p; scanf("%d", p);
C. int k, *p=&k; scanf("%d", p); D. int k, *p; *p= &k; scanf("%d" , p);

Section 2: Fill in the blanks (2 marks for each item, total 30 marks)

1. The following **for** loop statement will print out: **1 4 7 10 13 16 19 22 25**
- ```
for (i=1; i<=9; i++) printf("%3d", _____);
```
2. Given: **char c='@'**; , the value of expression '**A' <= c <= 'Z'**' is \_\_\_\_\_.
3. After executing the code below, the value of **y** is \_\_\_\_\_.
- ```
int x=0, y=0, z=0;
z = (x==1) && (y=2);
```
4. The value of expression **(double)(10/4*4)+2>7.0+4%5** is _____.
5. The output of calling **encode("2014\\")** is _____.
- ```
void encode(char *s)
{
 while (*s++) {
 switch(*s-'0') {
 case 0: case 2: case 4: case 8: printf("%c", *s+1); break;
 case 1: case 3: case 5: case 7: printf("%c", *s-1); break;
 default: printf("%c", *s);
 }
 }
}
```
6. The output of the code below is \_\_\_\_\_.
- ```
char s1[] = "Zhejiang";
char s2[] = "University";
char *s = (char*)malloc(strlen(s1)+strlen(s2)+1);
strcpy(s, s1);
strcat(s, s2);
*strchr(s, 'i') = 0;      /*strchr(s,c):return a pointer to the last instance of c in s*/
printf("%d#%s#",strlen(s),s);
```
7. Given: **char format[] = "No.%d%c"**; the statement **printf(format, 3, *("xyz"+2));** will print out _____.
8. Given the declaration: **int a[3][3]={1,2,3,4,5,6,7,8,9}**; ,the value of **a[-1][5]** is _____.

9. The output of the code below is _____:
- ```
int x=1,y=2,z=3;
while (x<z) printf("%d#", x+=z-=y);
```
10. The output of the code below is \_\_\_\_\_.
- ```
int k = 9;
int f(int x)
{
    static int k=0;
    k++;
    {
        int x = 0;
        return k+x;
    }
}
int main()
{
    while (k--=3) printf("%d#",f(k));
    return 0;
}
```
11. What are the values of *i,x,y* after executing the **do-while** loop? _____
- ```
int i,x,y;
i=x=y=0;
do {
 ++i;
 if (i%2) x+=i, i++;
 y +=i++;
} while (i<=7);
```
12. The following code fragment will print out \_\_\_\_\_.
- ```
int a[]={1,2,3,4,5,6,7}, *p, s;
for(s=0, p=a+1; p<=a+6; p=p+2) s+= *p;
printf("%d", s);
```
13. The symbolic constant which is usually used in **while ((c=getc(fp))!=_____)** as a mark of end-of-file.
14. The following code fragment will print out _____.
- ```
char s[]="abc\0xyz\0ghi", *sp=s;
printf("%s",sp+5);
```
15. The following function **strcat** will implement the task of concatenating(拼接) the string **t** to the end of string **s**.
- ```
void strcat(char *s, char *t)
{
    while (_____);
    while (*s++=*t++);
}
```

Section 3: Read each of the following programs and answer questions (5 marks for each item, total 30 marks)

1. When input:
This is a test {just for C},{You are} right!<ENTER>
the output of the following program will be _____.
- ```
#include <stdio.h>
main()
{
 char c;
 int state=0;

 while((c=getchar())!='\n') {
 if (c=='{' || c== '}')
```

```

 state= !state;
 else {
 if (state==1 && c>='a' && c<='z') c= c-'a'+'A';
 putchar(c);
 }
}
}

```

2. The following program will print out \_\_\_\_\_.

```

#include <stdio.h>
void sh(int array[], int n)
{
 int i, array_end;
 array_end = array[n-1];
 for(i=n-1; i>0; i--) array[i] = array[i-1];
 array[0] = array_end;
}
main()
{
 int number[]={0,1,2,3,4,5,6,7,8,9};
 int n= sizeof(number)/sizeof(int), m=3, i;
 for(i=0; i<m; i++) sh(number+m, n-m);
 for(i=0; i<n; i++) printf("%d#", number[i]);
}

```

3. When input: 123, the following program will print out \_\_\_\_\_.

```

#include <stdio.h>
void f(int n, char s[], int b)
{
 int i=0;
 while (n>0) {
 s[i++]=n%b+'0';
 n=n/b;
 }
 s[i]='\0';
}
main()
{
 char s[20];
 int n;

 scanf("%d",&n);
 f(n, s, 8);
 printf("%s\n", s);
}

```

4. The following program will print out \_\_\_\_\_.

```

#include <stdio.h>
int *p;
void f(int *po)
{
 static int s = 1;
 if(s > 1) {
 *po = *po + 1;
 if(*po == *p) printf("Yes ");
 else printf("No ");
 } else {
 s++;
 if(&po == &p) printf("Yes ");
 else printf("No ");
 }
}

```

```

 }
}
main()
{
 int a = 1;
 p = &a;
 f(p);
 printf("%d ", a);
 f(p);
 printf("%d", a);
}

```

5. The following program will print out \_\_\_\_\_.

```

#include <stdio.h>
void fun(char str[])
{
 int i,j;
 for (i=0,j=0;str[i];i++)
 if (str[i]>='0'&&str[i]<='9') str[j++]=str[i];
 str[j]='\0';
}
main()
{
 char str[100]=
 "By the end of 2013, the population of the #98 city has reached 765400.";
 fun(str);
 printf("%s\n",str);
}

```

6. Supposing: a text file "data.txt" contains some characters as follows:

13579-+QWERT{asdfs[]

After executing the following program, the contents of the file "data.txt" will include

\_\_\_\_\_.

```

#include <stdio.h>
#include <ctype.h>

int main()
{
 FILE *fp;
 int ch;
 long offset = 0L;
 if ((fp=fopen("data.txt", "r+"))==NULL) return -1;
 while (1) {
 fseek(fp, offset, SEEK_SET);
 /*fseek(FILE *fp, long offset, int start): locate the accessing position of a file.
 offset is the distance in bytes from the start point, SEEK_SET indicates that the
 start point is the beginning of the file. */
 if ((ch = getc(fp)) == '\n' || ch == EOF) break;
 if (ch>='A' && ch<='Z') {
 ch -= 'A' - 'a';
 } else if (ch>='a' && ch<='z') {
 ch -= 'a' - 'A';
 } else if (isdigit(ch)) {
 if (!isdigit(++ch)) ch -= 10;
 }
 fseek(fp, offset, SEEK_SET);
 putc(ch, fp);
 }
}

```

```

 offset++;
 }
 fclose(fp);
}

```

**Section 4: According to the specification, complete each program (2 marks for each blank, total 20 marks)**

1. The following program calculates the result of A+B. But the expression of A and B is different from the typical method. Each three digitals are separated by a ',' . For the following input:

-234,567,890 123,456,789

the program should get the result:

-111111101

You are supposed to complete the program.

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

```

int trim(_____ (1) _____)
{
 char *p = str;
 int flag = 1, num = 0;

 if (*p == '-') {
 flag = _____ (2) _____;
 p++;
 }
 while (*p != '\0') {
 if (*p != ',') num = _____ (3) _____;
 _____ (4) _____;
 }
 return num * flag;
}

```

```

main()
{
 char str1[20],str2[20];
 scanf("%s%s", str1, str2);
 printf("%d\n", trim(str1)+ _____ (5) _____);
}

```

2. There is a text file "dat.txt" which contains some integers with maximum number of 100. The following program will read in these integers, sort them in ascendant order(升序) and store them into another text file "sortdat.txt".

For example, suppose the file "dat.txt" contains the following numbers:

12 4 9 8 56 0 33 77 66 3 120 99

After executing the program, the file "sortdat.txt" will be listed as follows:

0 3 4 8 9 12 33 56 66 77 99 120

Fill in the blanks to complete the program.

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

```
#define MaxSize 100
```

```
FILE *OpenFile(char *filename, char *mode);
```

```
int ReadinNums(FILE *fp, int num[]);
```

```

void Sort(int num[], int n);
void SaveNums(FILE *fp, int num[], int n);

main()
{
 int num[MaxSize], n, i;
 _____ (6) _____;

 fpin = OpenFile("dat.txt", "r");
 fpout = OpenFile("sortdat.txt", "w");
 n = ReadinNums(fpin, num);
 Sort(num, n);
 SaveNums(fpout, num, n);
 fclose(fpin);
 fclose(fpout);
}

FILE *OpenFile(char *filename, char *mode)
{
 FILE *fp;

 if (_____ (7) _____ == NULL) {
 printf("Can't open file \"%s\"!\n", filename);
 exit(-1);
 }
 return fp;
}

int ReadinNums(FILE *fp, int num[])
{
 int count = 0;

 while (!feof(fp)) /*loop when not reach the end of file*/
 _____ (8) _____;

 return count;
}

void Sort(int num[], int n)
{
 int i, k, index, temp;

 for (i = 0; i < n-1; i++) {

```

```
index = i;
for (k = i+1; k < n; k++) {
 if (num[k] < num[index]) _____ (9) _____;
}
if (index != i) {
 temp = num[i];
 num[i] = num[index];
 num[index] = temp;
}
}
}
```

```
void SaveNums(FILE *fp, int num[], int n)
{
 int i;

 for (i = 0; i < n; i++) _____ (10) _____;
}
```