中山大学软件学院 **2010** 级软件工程专业 (2010 学年秋季学期)

《SS-122 程序设计(I)》期末试题 (B 卷)

(考试形式：闭　卷　　考试时间：2 小时)

警示

《中山大学授予学士学位工作细则》第六条

考试作弊不授予学士学位

方向：_____　　姓名：_____　　学号：_____

Section A: Multiple Choice (20 points, 2 points for each question)

For each of the following questions, select the best answer (only ONE) out of the choices list: (A), (B), (C), and (D).

1. Which of the following statements is FALSE?

(A) Machine languages are generally machine-dependent.

(B) People specify programs and data items as characters; computers then manipulate and process these characters as groups of zeros and ones.

(C) To refer to a particular location or element within an array, we specify the name of the array and the value of the particular element.

(D) The following are all valid variable names: _under_bar_, m928134, j7, his_account_total, a.

2. Suppose `float x = 4.5, y = 6.7; int a =8;`
what is the value of the expression: `x+a%3*(int)(x+y)%2/4`?

(A) 4　　(B) 4.5

(C) 5　　(D) 5.5

3. Which of the following C statements is correct?

(A) char hello[20] = "hello\\0";　　(B) scanf("%.2f", &y);

(C) (x+1)++;　　(D) char c; getchar(c);

4. Which operator has the associativity of "right to left"?

(A) &&　　(B) <=　　(C) ( )　　(D) ++

5. Which of the following C statements are wrong?

(A) char *str="I love China!";　str=str+7;　printf("%s\n", str);

(B) char str[14];　str = "I love China";

(C) char *str; str = "I love China";

(D) char str[14]="I love China";

6. How many times will "Hello" be printed by the following for loop?

```
int i,j;
for(i=0, j=5; ++i!=--j; )
```

```
printf("%s","Hello\n" );
```

(A) 6 times (B) 3 times

(C) Numerous times. It is a dead loop. (D) None of above

7.

```
struct card
{    char *face;
     char *suit;
 } a, *aPtr = &a;
```

After `a` is initialized, which of the following expressions CANNOT access the `face` component of `a` correctly?

(A) `a.face` (B) `aPtr->face`

(C) `(*aPtr).face` (D) `&a->face`

8. What will the value of `i` become after execute the following switch statement?

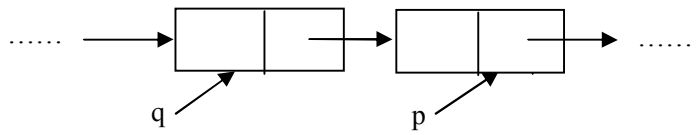
```
int i=20;
switch(i)
{
  case 10: i+=1;
  case 20: i+=2;
  case 30:i+=3;
  default: i-=1;
}
```

(A) 22 (B) 23 (C) 24 (D) 25

9. Suppose we have the following node definition and declarations:

```
struct node
{   int key;
    struct node *next;
}*head, *p, *q, *new;
```

Suppose `new` has been initialized; `p` and `q` are pointing to two adjacent nodes of a linked list, as shown below:



Which of the following statement is CORRECT to insert the node new between p and q?

(A) `q->next = new;  new->next = p->next;`

(B) `new->next = p; q->next = new;`

(C) `*q->next = new; *new->next = p;`

(D) `(*q).next = *new; (*new).next = *p;`

10. For `int a[3][4], (*p)[4]=a;` Which of the following CANNOT access `a[i][j]` correctly?

(A) `*a[i]+j`
(B) `(*(p+i))[j]`
(C) `p[i][j]`
(D) `*(*p+4*i+j)`

Section B: Short Answer (40 points)

Just briefly answer the questions according the requirements.

1. (7 points) C language has three kinds of control statements, including: 1 of sequence statement, 3 of selection statements, and 3 of repetition statements.
   Please draw the corresponding flow charts for each of the 7 control statements.

2. (8 points) Can the following program be complied and executed correctly?
   If no, please explain why; else, please write down the printed content.

```
#include <stdio.h>
void staticArray( void )
{  static int x[3] = {1};
   int i=0;
   for (; i <= 2; i++ )
      printf( "x[%d] = %d  ", i, x[i] += 5 );
   printf("\n");
}
int x =10;
int main()
{
   int y = 2;
  {    int x = 5;
       x -= y;
  }
  x -= y;
  printf( "x = %d\n", x);
  staticArray();
  staticArray();

  return 0;
}
```

3. (8 points) Given an array that contains **double** values, write a function **void** `shiftRight(double a[], int size, int k)` that shifts the array elements k positions to the right. The first `k` elements are replaced by **zero**.

Here suppose `k` is non-negative and it is smaller than the size of the array. Your program should return and show a message if `k` is negative or larger than the size of the array, and your program must not use any additional array for shifting.
For example, if `a[]={1.2, 3.4, 1.8, 5.0, 8.8, 2.1, 1.1}`, then after shifting right 3 positions, the array `a` becomes `{0.0, 0.0, 0.0, 1.2, 3.4, 1.8, 5.0}`.

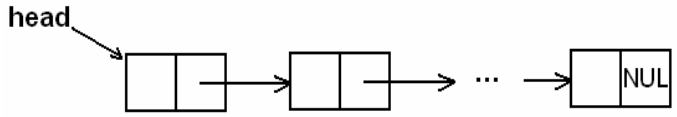
4. (6 points) What does the following program do?

```
#include <stdio.h>

int mystery( unsigned bits );

int main()
{
   unsigned x;
   printf( "Enter an integer: " );
   scanf( "%u", &x );
   printf( "The result is %d\n", mystery( x ) );
   return 0;
}

int mystery( unsigned bits )
{
   unsigned i;
   unsigned mask = 1 << 31;
   unsigned total = 0;

   for ( i = 1; i <= 32; i++, bits <<= 1 )
  {
      if ( ( bits & mask ) == mask )
          total++;
  }
  return !( total % 2 ) ? 1 : 0;
 }
```

提示: Enter 65, the result is 1; enter 7, the result is 0.

5. (6 points) A singly-linked list with head node consists of a head node pointing to the sequence of data, and in each data record there is a field containing a link to the next record in the sequence, as shown below:

```
typedef struct node
{   int key;
    struct node *next;
}Node;
```



The following function double `GetLength(Node *)` is to find the length of a singly-linked list. Please write ONE C statement in each blank to complete the function definition.

```
int GetLength(Node *head)
{
    int  Len;
    _____(1)_____

    while ( _____(2)_____ )
    {    Len++;
         _____(3)_____
    }
    return Len;
}
```

6. (5 points) Following three functions are written by your classmates, or maybe by yourself. They are of the same function: to judge if the input integer is a prime number (素数), if yes, return 1, otherwise, return 0.
Are they correct? Comparing the three pieces of C codes, which one is the best? Why?

```
/*written by student A*/
int isPrime(int n)
{
  int i;
  for(i=2;i<sqrt(n);i++)
    {
       if(n%i==0)
            return 0;
    }
    return 1;
}
```

```
/*written by student B*/
int susu(int n)
{ int i, flag=1;
  for(i=2;i<sqrt(n);i++)
  if(n%i==0)
  flag=0;
  return flag;
}
```

```
/*written by student C*/
int isPrim(int n)
{
  int i, count=0;
  for(i=2;i<=n;i++)
      if(n%i==0)
          count++;
  if(count == 2)
       return 1;
  else
       return 0;
}
```

Section C: Program Output Analysis (20 points)

Write the result after executing the following programs or program fragments.

1. (9 points) Given the following program, what is the output?

```
#include <stdio.h>
int aMethod(int *a, int *b)
{   *a -= *b;
    *b += *a;
    return *a**b;
}

int main()
{   int a = 2, b = 3, c=0;
    a = aMethod(&a, &b);
    aMethod(&b, &c);
    b = aMethod(&a, &c);
    printf("%d, %d, %d\n", a, b, c);
    return 0;
}
```

2. (6 points) Given the following program, what is the output?

```
int f(int n)
{    return 2 * n;
}

int f2(void)
{   int count = 0, k =0;
    for (; k<90; k+=3)
    {
        if (k%9 == 0) continue;
        if (k%10 == 0) break;
        count++;
    }
    return count;
}
int main(int argc, char *argv[])
{
   printf("%d, %d\n",  f(f(f(5))), f2());
   return 0;
}
```

3. (5 points) Given the following program, what is the output if "12345\n" is inputted?

```c
#include <stdio.h>

void myPrint(void)
{   int c;
    if ( (c=getchar())!='\n')
    {
        myPrint();
        putchar(c);
    }
    putchar('\n');
}
int main()
{   myPrint();
    return 0;
}
```

Section D: Program Error Correction (20 points)

1. (10 points) The following program simulates 6000 rolls of a dice (骰子) to show that each face of the die shows up with approximately equal probability. The program uses an int array `freq` to store the frequency of each face of a die. 7 elements are declared for the array, but we use only freq[1] to freq[6] for the six faces and leave freq[0] unused.
The program contains errors. Please study the program and identify all the errors. You should indicate the line number where the error occurs and provide the corrected version. For omission, you should indicate what the omission is and where it should appear.
Function **int** `rand(`**void**`)` returns a pseudo-random integer between 0 to RAND_MAX.

```
1:  #include <stdio.h>;
2:  #include <stdlib.h>
3:
4:  int main()
5:  {
6:    int freq[7], i, face;
7:    for ( i = 1; i < 6000; i++ )
8:    {
9:        face = rand();
10:        ++ freq[face];
11:   printf( "%s%13s\n", "Face", "Frequency" );
12:   for(i=1;  i<=6 ; i++)
13:       printf( "%4d%13d\n",  i, freq[i] );
14:   return 0;
15: }
```

2. (10 points)The following program is supposed to using Bubble Sort method to sort the array a in descending order, and then print out all the elements in a. However, the following version of this program is severely broken. Please fix it.

```
1:  #include <stdio.h>
2:  int main()
3:  {   double a[]= {0.1, -2.3, 3.4, -5.6, 9.0};
4:  int i, n = sizeof(a);
5:     bubbleSort(a, n);
6:  printf("The array a after bubble sorting becomes:\n");
7:     for(i=0;i<n;i++)
8:        printf("%d\t", a[i]);
9:     return 0;
10: }
11:
12: void bubbleSort(double *array, const int size)
13: {  int i, pass;
14:
15:    for( pass = 1; pass < size; pass++ )
16:       for( i = 0; i < size - pass; i++ )
17:          if( *(array+i) < *(array+i+1) )
18:          {
19:                *(array+i) = *(array+i+1);
20:                *(array+i+1) = *(array+i);
21:           }
22: }
```