
Q1 The following recursive function takes two non-negative integers m and n as arguments and recursively computes and returns the product mn using the formula mn = (m-1)n + n. Fill in the blank involving an appropriate recursive call. You are not allowed to use the multiplication operator (*) anywhere in the function.

```
int f ( int m, int n )
{
   if (m == 0) return 0;
   return __________ ;
}
```

ANS: n + f(m-1,n)

Q2 Fill in the blank so that the following recursive function returns the average of n numbers stored in an integer array A.

```
      float  ComputeAvg (int *A, int n)
      {
         float avg;

[A]      if (n == 1) return ( __________ );
[B]      avg = (A[0] + __________ ) / n ;  /* Make a recursive call */
         return avg;
      }
```

ANS:

[A] (float)A[0]

[B] (n-1) * ComputeAvg(&A[1], n-1)

Q3 What should be the code in the blank space below so that func(351274) prints 47215351274?

```
void func (int n)
{
   if (n < 10) printf("%d",n);
   else {
      printf("%d", n%10);
      __________   /* Recursive call of func() */
      printf("%d", n%10);
   }
}
```

ANS: func(n/10);

Q4 Let A be an n x n matrix with entries a_{ij}. We want to write A as a sum of an upper-triangular matrix B (a matrix having $b_{ij} = 0$ if $i > j$) and a lower-triangular matrix C (a matrix having $c_{ij} = 0$ if $i \leqslant j$). Fill in the following blank to achieve this task. Assume that $n \leqslant 100$.

```
int A[100][100], B[100][100], C[100][100];

for (i=0; i<n; ++i) {
   for (j=0; j<n; ++j) {
      __________
      __________
```

```
      }
   }
```

ANS:

```
   B[i][j] = (i > j) ? 0 : A[i][j];
   C[i][j] = (i <= j) ? 0 : A[i][j];
```

Q5 Consider the following declaration of a 2-dimensional array x.

```
   int x[4][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12}};
```

Assume that the starting address of x is 2000 (in decimal), and an int variable requires four bytes of memory. What are the values (in decimal) of x+1 and *(x+2)+3?

ANS: 2012 and 2036

Q6 Consider an array A[10] of type _CITY (as defined below). The member city will store a null-terminated string representing the name of a city.

```
   typedef struct {
      char city[20];
      int population;
   } _CITY;

   _CITY A[10];
```

Fill in the blank in the following code fragment that checks if there are any two cities with the same name in the array.

```
   for (i=0; i<9; i++)
      for (j=i+1; j<10; j++)
         if ( __________ )
            printf("There are two cities with the same name\n");
```

ANS: strcmp(A[i].city, A[j].city) == 0

Q7 Consider a variable X of type struct T2 (as defined below).

```
   struct T1 {
      float A[10];
   };

   struct T2 {
      struct T1 B[20];
   };

   struct T2 X;
```

Fill in the blank in the following code fragment so that it prints all float values in all struct T1 type variables inside X in a single line.

```
for (i=0; i<20; i++)
   for (j=0; j<10, j++)
      printf( __________ );
```

[ANS] "%f ", X.B[i].A[j]

Q8 Let p point to an array of n integers. We want to duplicate each element of this array, and store the result in an array pointed to by q. For example, if the array pointed to by p stores 3, 1, 2, 2, 6, then the array pointed to by q should store 3, 3, 1, 1, 2, 2, 2, 2, 6, 6. Fill in the blank to complete the code of the following function.

```
   void duplicate ( int *p, int n, int *q )
   {
      int i;

      for (i=0; i<n; ++i) {
         __________
         q += 2; ++p;
      }
   }
```

ANS:

```
   *q = *(q+1) = *p;
```

or

```
   q[0] = q[1] = p[0];
```

Q9 Consider an array A of size 5 storing positive integers only. If the value of the variable x after executing the following program fragment is 20, what can be the possible initial values of the integers in A?

```
   x = 0;
   for(i=0; i<5; i++)
      x += (*A + i) + *(A + i);
```

ANS: {1, 1, 1, 1, 1}

Q10 Fill in the blanks so that the following function F takes a null-terminated string S as a parameter and returns (through parameters) the first non-digit character in S. Assume that there is at least one non-digit character in S. You cannot change the return type from void.

```
[A]   void F ( __________ )
      {
         int i;

         for (i=0; S[i] != '\0'; i++) {
            if (S[i] < '0' || S[i] > '9') {
[B]            __________
               return;
            }
         }
      }
```

ANS:

```
   [A] char *S, char *ch
   [B] *ch = S[i];
```
