

CS11002 Programming and Data Structures, Spring 2007–08

Class test 2

Maximum marks: 20

March 02, 2008

Total time: 1 hour

Roll no: _____ Name: _____ Section: _____

[Write your answers in the question paper itself. Submit the supplementary sheet(s) only if used for your final answer. Be brief and precise. Answer all questions.]

1. Write a function that accepts as argument an array A of integers together with its size n and a non-negative integer k . The function should return another array, allocated dynamically within the function, which is obtained by cyclically shifting the input array A by k positions to the right. For example, upon the input of $A = \{2, 4, 6, 1, 3, 9, 5\}$ of size $n = 7$ and $k = 3$, your function should return $\{3, 9, 5, 2, 4, 6, 1\}$. (6)

```
int *rotateArray ( int A[], unsigned int n, unsigned int k )  
{
```

```
int *p, i, j;  
  
p = (int *)malloc(n * sizeof(int));  
for (i=0, j=k%n; i<n; ++i) {  
    /* Index i is for reading and index j is for writing */  
    p[j] = A[i];  
    if (++j == n) j = 0;  
}  
return p;
```

```
}
```

2. (a) Define a data type **vector** which consists of a dynamically allocated array to store the elements of a vector and an integer to designate the size (dimension) of the vector. Assume that we deal with vectors of floating-point numbers. (2)

```
typedef struct {  
    unsigned int dim;  
    double *element;  
} vector;
```

(b) Write a function that, for two input vectors $\mathbf{v} = (v_0, v_1, \dots, v_{n-1})$ and $\mathbf{w} = (w_0, w_1, \dots, w_{n-1})$, returns the dot product $\mathbf{v} \cdot \mathbf{w} = v_0w_0 + v_1w_1 + \dots + v_{n-1}w_{n-1}$. If the input vectors are not of the same dimension, your function should return some appropriate error value. (For example, the macro `HUGE_VAL` defined in `math.h` translates to `Inf` in a machine supporting IEEE-754 arithmetic.) (6)

```
double dotProduct ( vector v, vector w )
{
```

```
double sum = 0.0;
unsigned int i;

if (v.dim != w.dim) return HUGE_VAL;
for (i=0; i<v.dim; ++i) sum += v.element[i] * w.element[i];
return sum;
```

```
}
```

3. A two-dimensional character array is used to store a list of names. Each name is stored in a row. An empty string indicates the end of the list. Complete the *recursive* function `printNames()` to print the names stored in the two-dimensional array supplied as `p`. (6)

```
void printNames ( char (*p)[100] )
{
```

```
if (p[0][0] == '\0') return;
printf("%s\n", *p);
printNames(p+1);
```

```
}
```

```
int main()
{
    char names[20][100] = { "Bombay", "Delhi", "Guwahati", "Kanpur",
                           "Kharagpur", "Madras", "Roorkee", "" };

    printNames(names);
    return 0;
}
```