Indian Institute of Technology, Kharagpur

CS19001 Programming and Data Structures Laboratory, Autumn 2019

Assignment for Week 6 (September 30, 2019)

Total Marks: 40 **Submission Deadline:** 17:45

INSTRUCTIONS

1. Submit a separate C file for each of the problems. The solution for problem i should be named **[rollno]-prob$i$.c** where '[rollno]' is your roll number.
2. You may consult your notes, books or manual pages.

PROBLEMS

1. A *multivariate polynomial* of degree d in n variables $x_1, \ldots, x_n$ is of the form

$$\sum_{k \leq d} c_k x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n},$$

where $k = \sum_{i=1}^{n} k_i$ represents the degree of the term with coefficient c_k. Each coefficient monomial combination is called a term. For example, $4a^4b^3d^2 - 10b^2cdf^3 + abd^2e + 7cef$ is a multivariate polynomial in variables a, b, c, d, e, f containing four terms – $4a^4b^3d^2, -10b^2cdf^3, abd^2e, 7cef$ of degrees 9, 7, 5, 3 respectively.

Define two structures `term` and `mpoly` as follows. Let `maxVar` and `maxTerms` denote the maximum number of variables and maximum number of terms in a polynomial respectively. Set `maxVar` to 26 so that you can use distinct lowercase letters (a,b,. . . ,z) to represent each variable. Assume all coefficients are integers.

```
struct term{
  int coeff;       //stores coefficient
  int exp[maxVar]; //stores exponents corresponding of each variable
};

struct mpoly{
  int nTerms;                 // number of terms -- to be read from the user
  int nVar;                   // number of variables
  struct term poly[maxTerms]; // array of structures of type term
};
```

Write functions to

- Read a polynomial from the user
- Display the polynomial

For example, when the polynomial $4a^4b^3d^2 - 10b^2cdf^3 + abd^2e + 7cef$ is read into a variable `p` of type `struct mpoly`, then the variable `p.nTerms` will be 4 and `p.nVar` would be 6. The terms are stored in `p.poly[0], p.poly[1], p.poly[2], p.poly[3]`.

Consider the first term – its coefficient is 4 and so `p.poly[0].coeff = 4`. As there are 6 variables fixed to first 6 letters of the alphabet we have `p.poly[0].exp[0,1,2,3,4,5] =` {`4,3,0,2,0,0`} i.e., `exp[i]` indicates

the exponent of variable numbered `i+1` or equivalently variable represented by the character `'a'+ (char) i`. Marks: 20

Practice Question (not for evaluation): Write functions to add and multiply multivariate polynomials.

Sample Output

```
Enter number of variables: 6
Variables named a,b,c,d,e,f

Enter number of terms: 4
Input the terms (in variables a,b,c,d,e,f) in decreasing order of degree

Reading term 1...
Coefficient: 4
Exponent of a: 4
Exponent of b: 3
Exponent of c: 0
Exponent of d: 2
Exponent of e: 0
Exponent of f: 0

Reading term 2...
Coefficient: -10
Exponent of a: 0
Exponent of b: 2
Exponent of c: 1
Exponent of d: 1
Exponent of e: 0
Exponent of f: 3

Reading term 3...
Coefficient: 1
Exponent of a: 1
Exponent of b: 1
Exponent of c: 0
Exponent of d: 2
Exponent of e: 1
Exponent of f: 0

Reading term 4...
Coefficient: 7
Exponent of a: 0
Exponent of b: 0
Exponent of c: 1
Exponent of d: 0
Exponent of e: 1
Exponent of f: 1

The input polynomial is

4 a^4 b^3 d^2 - 10 b^2 c d f^3 + a b d^2 e + 7 c e f
```

2. Consider the following code.

```
#include <stdio.h>
#define MAX_SIZE 25

int main(){
  int size, i, j;
  int **array;
  printf("Enter the size (<= %d) of the square matrix: ", MAX_SIZE);
  scanf("%d", &size);

  // check that size < MAX_SIZE
  // allocate (size x size) memory for array
  // fill array

  for(i=0; i<size; i++){
    for(j=0; j<size; j++){
      printf("%5d ", array[i][j]);
    }
    printf("\n");
  }
  return 0;
}
```

Your task is to replace the commented lines with code that fills `array[][]` with entries that match the desired output. For example, if the desired output is of the form

```
 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
```

for `size = 5`, then the code replacing the commented line `// fill array` will be

```
for(i=0; i<size; i++){
  for(j=0; j<size; j++){
    a[i][j] = i*size + j + 1;
  }
}
```

Write code for generating the following patterns.

(a)

```
 1   2   3   4   5
10   6   7   8   9
14  15  11  12  13
18  19  20  16  17
22  23  24  25  21
```

Marks: 5

(b)

```
 5   4   3   2   1
10   9   8   7   6
15  14  13  12  11
20  19  18  17  16
25  24  23  22  21
```

Marks: 5

(c)

```
 1   2   3   4   5
16  17  18  19   6
15  24  25  20   7
14  23  22  21   8
13  12  11  10   9
```

Marks: 10

Note that the user enters `size`. Your code should work for any size $\leq$ 25 (not just 5). Submit a single C file. You may ask the user to enter a choice (a,b,c) to choose the pattern.