# ASSIGNMENT 1 REPORT

# CE1007 DATA STRUCTURES

**EDWIN CANDINEGARA (FE2)**

**U1320135K**

**SEMESTER 2**

**AY 2013/2014**

**SCHOOL OF COMPUTER ENGINEERING**

**NANYANG TECHNOLOGICAL UNIVERISTY**

## Full Code:

```
/*
       Author / Lab Group: Edwin Candinegara / FE2
       Program name: FE2 Edwin Candinegara.c
       Date: 02 March 2014
       Purpose: Implementing the required functions for Assignment 1
*/

/* Preprocessor Instructions */
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
#include <math.h>

/* Constants Declaration */
#define SIZE 4
#define SIZE1 5
#define SIZE2 10

/* Structure Declaration */
struct student{
    char name[20]; /* student name */
    double testScore; /* test score */
    double examScore; /* exam score */
    double total; /* total score = (test + exam scores) / 2 */
};

typedef struct {
    double x;
    double y;
} Point;

typedef struct {
    Point topLeft; /* top left point of rectangle */
    Point botRight; /* bottom right point of rectangle */
} Rectangle;
```

```c
/* Function Prototypes */
void readMatrix(int matrix[SIZE][SIZE]);
void displayMatrix(int matrix[SIZE][SIZE]);
void computeTotal(int matrix[SIZE][SIZE]);
void compress(char data[5][10]);
void squeeze(char str[], char c);
char *strrchr2(char *s, char ch);
void findWord(char word[][20], char *first, char*last);
double computeArea(Rectangle *r);
void computeScore();
int fun(int n);
int countZeros(int num);
void reverseAr(char ar[], int n);


/* Main Program */
int main() {

    /* Variable declaration */
    int choice;  /* For choosing the menu */
    int i, j;  /* Loop counters */
    int n;  /* Store integer in question 8 and 9 */
    int zeros;  /* Store the number of zeros in question 9 */
    int arFunc1[SIZE][SIZE];  /* int array for question 1 */
    char str[200];  /* String for question 3, 4, and 10 -> reduce the number of
                       variables needed */
    char strFunc2[5][10];  /* String for question 2 */
    char strFunc5[5][20];  /* Array of string for question 5 */
    char c, first[20], last[20];  /* c: target character for question 3 and 4; first,
                                     last: string for question 5 */
    char *ptrStr4;  /* String pointer for question 4 */
    double recArea;  /* To store the rectangle area in question 6 */
    Rectangle structureFunc5;  /* Structure containing the two points of the rectangle for
                                  question 6 */


    /* Print menu */
    printf("\nPerform the following functions ITERATIVELY:\n");
    printf("1: computeTotal()\n");
    printf("2: compress()\n");
    printf("3: squeeze()\n");
    printf("4: strrchr2()\n");
    printf("5: findWord()\n");
    printf("6: computeArea()\n");
    printf("7: computeScore()\n");
    printf("8: fun()\n");
    printf("9: countZeros()\n");
    printf("10: reverseAr()\n");
    printf("11: quit\n");

    do {
        /* Choose which function to be run */
        printf("\nEnter your choice: ");
        fflush(stdin);
        scanf("%d", &choice);

        /* Run the chosen function */
        switch (choice) {

               /* Question 1 */
               case 1 :
                      /* Take a 4x4 matrix as an input */
                      readMatrix(arFunc1);

                      /* Change the last column into the sum of the first three columns */
                      computeTotal(arFunc1);
                      break;

               /* Question 2 */
               case 2 :
                      printf("Enter your data (5x10 characters):\n");

                      /* Loop to take the 5x10 characters for the array */
                      for (i = 0; i < SIZE1; i++) {
                            /* Prevent any escape character to be considered as the
                            next sub-array input */
                            fflush(stdin);
                            for (j = 0; j < SIZE2; j++)
```

```
            scanf("%c", &strFunc2[i][j]);
    }

    /* Compress the array of characters */
    compress(strFunc2);
    break;

/* Question 3 */
case 3 :
    printf("Enter a string: ");
    fflush(stdin);
    gets(str);

    printf("Enter a char: ");
    fflush(stdin);
    scanf("%c", &c);

    /* Remove all the targeted characters in the string */
    squeeze(str, c);
    printf("Squeezed String: %s\n", str);
    break;

/* Question 4 */
case 4 :
    printf("Enter a string: ");
    fflush(stdin);
    gets(str);

    printf("Enter the target char in the string: ");
    fflush(stdin);
    scanf("%c", &c);

    /* Search and store the address of the last target character
    found */
    ptrStr4 = strrchr2(str, c);
    printf("Resultant string: %s\n", ptrStr4);
    break;
```

```c
/* Question 5 */
case 5 :
        printf("Enter 5 words separated by space: ");
        fflush(stdin);

        /* Take inputs of string */
        for (i = 0; i < SIZE1; i++)
                scanf("%s", strFunc5[i]);

        /* Find the first and the last word of the string array */
        findWord(strFunc5, first, last);
        printf("First word: %s, Last word: %s\n", first, last);
        break;

/* Question 6 */
case 6 :
        printf("Enter top left point: ");
        scanf("%lf %lf", &structureFunc5.topLeft.x, &structureFunc5.topLeft.y);

        printf("Enter the bottom right point: ");
        scanf("%lf %lf", &structureFunc5.botRight.x,structureFunc5.botRight.y);

        /* Calculate and store the rectangle’s area */
        recArea = computeArea(&structureFunc5);
        printf("Area = %lf\n", recArea);
        break;

/* Question 7 */
case 7 :
        /* Create a database of maximum 50 students */
        computeScore();
        break;

/* Question 8 */
case 8 :
        printf("Enter a number: ");
        fflush(stdin);
        scanf("%d", &n);
```

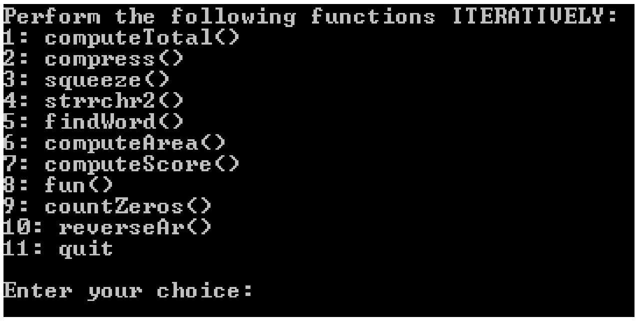
```c
                        /* Print the result of fun(n) */
                        printf("Result: %d\n", fun(n));
                        break;

                /* Question 9 */
                case 9 :
                        printf("Enter a number: ");
                        fflush(stdin);
                        scanf("%d", &n);

                        /* Count the number of zeros in the integer */
                        zeros = countZeros(n);
                        printf("Number of zeros: %d\n", zeros);
                        break;

                /* Question 10 */
                case 10 :
                        /* Even though it is an array of characters, I process it just like
                        using string data type */
                        printf("Enter an array of characters: ");
                        fflush(stdin);
                        gets(str);

                        /* Reverse the string */
                        reverseAr(str, strlen(str));
                        printf("The reversed array of characters: %s\n", str);
                        break;
        }
    } while (choice < 11);

    return 0;
}

/* Functions Codes */
void readMatrix(int matrix[SIZE][SIZE])
{
    int i,j;
    /* Take a 4x4 matrix as input */
    printf("Enter matrix (4x4): \n");
    for (i=0; i<SIZE; i++)
        for (j=0; j<SIZE; j++)
                scanf("%d", &matrix[i][j]);

    printf("\n");
}

void displayMatrix(int matrix[SIZE][SIZE])
{
    int i,j;

    /* Print the 4x4 matrix */
    printf("The resulting matrix (4x4): \n");
    for (i = 0; i < SIZE; i++) {
        for (j = 0; j < SIZE; j++)
                printf("%d ", matrix[i][j]);

        printf("\n");
    }

    printf("\n");
}

/* Question 1 */
void computeTotal(int matrix[SIZE][SIZE])
{
    int r, c;

    /* Loop to change directly the rightmost column as the sum of the first until third
    column */
    for (r = 0; r < SIZE; r++) {
        matrix[r][SIZE - 1] = 0;
        for (c = 0; c < SIZE - 1; c++)
                /* Change the rightmost column to the sum of the previous columns */
                matrix[r][SIZE - 1] += matrix[r][c];
        }
        /* Print the resulting matrix */
        displayMatrix(matrix);
}

/* Question 2 */
void compress(char data[5][10]) {
    int count, r, c;
    char store;

    printf("\nThe compression output:\n");
    for (r = 0; r < SIZE1; r++) {
        store = data[r][0];
        count = 0;

        /* Loop for comparing a character with the next character */
        for (c = 0; c < SIZE2; c++) {
                if (data[r][c] == store)
                        count++;
                else {
                        /* When the character is different, it prints the previous character
                        with the count and also saves the new different character */
                        printf("%c%d", store, count);
                        store = data[r][c];
                        count = 1;
                }
        }

        /* Print the very last checked character */
        printf("%c%d\n", store, count);
    }
}

/* Question 3 */
void squeeze(char str[], char c) {
    int i = 0, count = 0;

    /* Loop to check and remove the character c inside the string */
    while (str[i]) {
```

```c
        /*  Count is only increased if str[i] != c such that if str[i] == c, then the next
        characters position will be moved backward */
        if (str[i] != c) {
                str[count] = str[i];
                count++;
        }

        i++;
    }

    /* Give a NULL after the last character of the squeezed string */
    str[count] = NULL;
}

/* Question 4 */
char *strrchr2(char *s, char ch) {
    char *ptr = NULL;
    int i = 0, count = 0;

    /* Loop for looking the last target character location in the string*/
    while (*(s + i)) {
        if (*(s + i) == ch) {
                /* Assign ptr to the location where the target character is found */
                ptr = (s + i);
                count++;
        }

        i++;
    }

    return ptr;
}

/* Question 5 */
void findWord(char word[][20], char *first, char *last) {
    int i, compare1, compare2;

    /* Preparing for the comparison */
    strcpy(first, word[0]);
    strcpy(last, word[0]);

    /* Loop for comparing one word with the first and last word */
    for (i = 0; i < SIZE1; i++) {
        /* Comparing */
        compare1 = strcmp(first, word[i]);
        compare2 = strcmp(last, word[i]);

        /* Each word can only go to one category of the if block */
        if (compare1 > 0)
                strcpy(first, word[i]); /* Change the first word */
        else if (compare2 < 0)
                strcpy(last, word[i]); /* Change the last word */
    }
}

/* Question 6 */
double computeArea(Rectangle *r) {
    /* Print the two points */
    printf("Top left x: %lf y: %lf\n", r->topLeft.x, r->topLeft.y);
    printf("Bottom right x: %lf y: %lf\n", r->botRight.x, r->botRight.y);

    /* Return the area */
    return fabs((r->botRight.x - r->topLeft.x) * (r->topLeft.y - r->botRight.y));
}

/* Question 7 */
void computeScore() {
    struct student record[50];
    int i, j = 0;
    double sum = 0;

    /* The maximum number of students is 50 */
    for (i = 0; i < 50; i++) {
        /* Input student name */
        printf("Enter student name: ");
        fflush(stdin);
        gets(record[i].name);
        /* Go out of the loop if the name input is "END" */
        if (strcmp(record[i].name, "END") == 0)
            break;

        /* Test Score */
        printf("Enter test score: ");
        fflush(stdin);
        scanf("%lf", &record[i].testScore);

        /* Exam Score */
        printf("Enter exam score: ");
        fflush(stdin);
        scanf("%lf", &record[i].examScore);

        /* Total score */
        record[i].total = (record[i].testScore + record[i].examScore) / 2;

        /* Loop for locating the first white space from the student’s name */
        while (record[i].name[j] != 32 && record[i].name[j] != NULL)
            j++;

        /* Giving NULL at the end of the first name */
        record[i].name[j] = NULL;

        /* Sum of all total score */
        sum += record[i].total;
        printf("Student %s total: %lf\n\n", record[i].name, record[i].total);

        /* Reset the counter j */
        j = 0;
    }

    printf("Overall average: %lf\n", sum / i);
}

/* Question 8 */
int fun(int n) {
    /* Formula is based on the manual */
    if (n <= 1)
```

```
        return 1;
    else if (n % 2 == 0)
        return fun(n / 2);
    else
        return 2 * fun((n - 1) / 3);
}

/* Question 9 */
int countZeros(int n) {
    if (n < 10)
        /* Return 0 except if n == 0 -> check using short conditional statement */
        return (n == 0)? 1 : 0;

    else
        /* Use short conditional statement to return */
        return (n % 10 == 0)? 1 + countZeros(n / 10) : countZeros(n / 10);
}

/* Question 10 */
void reverseAr(char ar[], int n) {
    char temp;
    int length = strlen(ar);

    /* Swap the first char with the last char, second char with the second last char
    and so on without interrupting the NULL character  */
    if (n == length / 2)
        return;
    else {
        temp = ar[length - n];
        ar[length - n] = ar[n - 1];
        ar[n - 1] = temp;

        /* Call itself */
        reverseAr(ar, (n-1));
        return;
    }
}
```

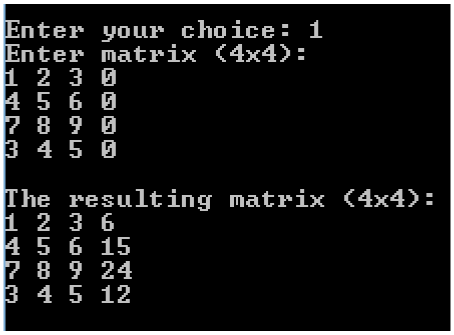
## Results of the code:

### Menu:



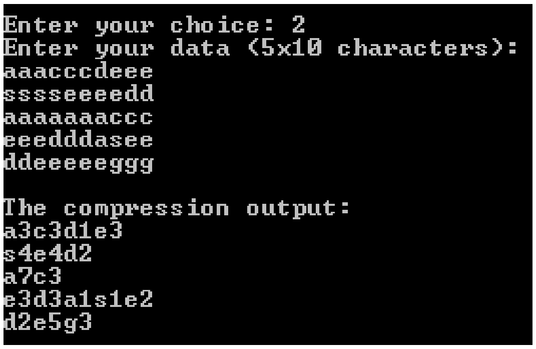
```
Perform the following functions ITERATIVELY:
1: computeTotal()
2: compress()
3: squeeze()
4: strrchr2()
5: findWord()
6: computeArea()
7: computeScore()
8: fun()
9: countZeros()
10: reverseAr()
11: quit

Enter your choice:
```

1. **Question 1**



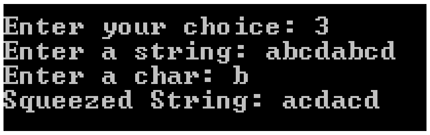
```
Enter your choice: 1
Enter matrix (4x4):
1 2 3 0
4 5 6 0
7 8 9 0
3 4 5 0

The resulting matrix (4x4):
1 2 3 6
4 5 6 15
7 8 9 24
3 4 5 12
```

2. **Question 2**



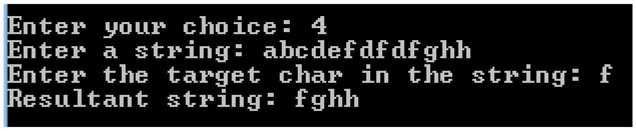
```
Enter your choice: 2
Enter your data (5x10 characters):
aaacccdeee
sssseeeedd
aaaaaaaccc
eeedddasee
ddeeeeeggg

The compression output:
a3c3d1e3
s4e4d2
a7c3
e3d3a1s1e2
d2e5g3
```

3. **Question 3**



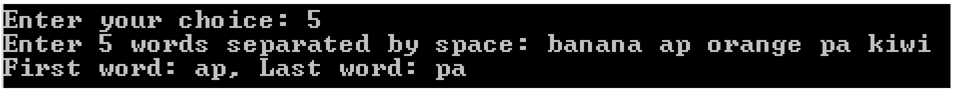
```
Enter your choice: 3
Enter a string: abcdabcd
Enter a char: b
Squeezed String: acdacd
```

## 4. Question 4



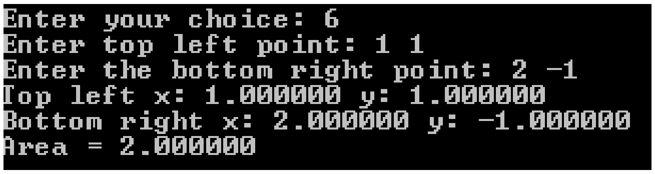
```
Enter your choice: 4
Enter a string: abcdefdfdfghh
Enter the target char in the string: f
Resultant string: fghh
```

## 5. Question 5



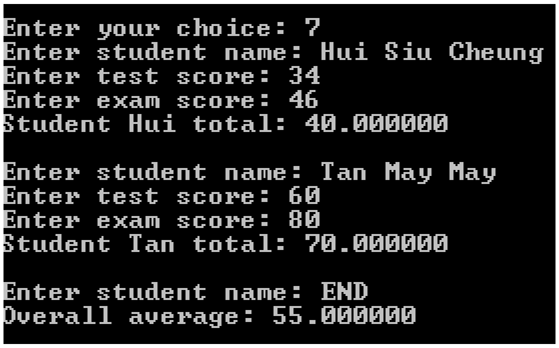
```
Enter your choice: 5
Enter 5 words separated by space: banana ap orange pa kiwi
First word: ap, Last word: pa
```

## 6. Question 6



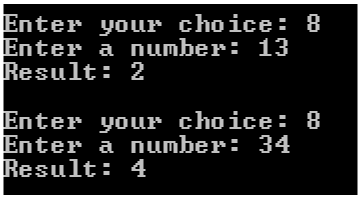
```
Enter your choice: 6
Enter top left point: 1 1
Enter the bottom right point: 2 -1
Top left x: 1.000000 y: 1.000000
Bottom right x: 2.000000 y: -1.000000
Area = 2.000000
```

## 7. Question 7



```
Enter your choice: 7
Enter student name: Hui Siu Cheung
Enter test score: 34
Enter exam score: 46
Student Hui total: 40.000000

Enter student name: Tan May May
Enter test score: 60
Enter exam score: 80
Student Tan total: 70.000000

Enter student name: END
Overall average: 55.000000
```

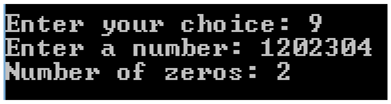
## 8. Question 8



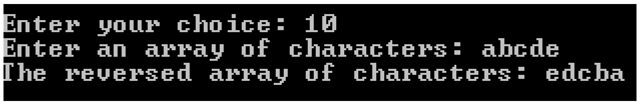
```
Enter your choice: 8
Enter a number: 13
Result: 2

Enter your choice: 8
Enter a number: 34
Result: 4
```

## 9. Question 9

10.



```
Enter your choice: 9
Enter a number: 1202304
Number of zeros: 2
```



```
Enter your choice: 10
Enter an array of characters: abcde
The reversed array of characters: edcba
```