

C Assignment 3

Name:

Student ID:

Exercise Questions

1. (4 marks) See the following two statements.

```
int a[3] = {11, 22, 33};  
int *pa = a;
```

Give the values of the following expressions.

*a =

*(a+2) =

*pa =

pa[1] =

2. (6 marks) See the following two statements.

```
int m[4][4] = {{1,3,5,7}, {11,33,55,77}, {2,4,6,8},  
{22,44,66,88}};  
int (*parr)[4] = m;
```

Give the values of the following expressions.

**m =

*(*m+2) =

*(*(m+1)+1) =

*(m[1]+2) =

(*(m+2))[3] =

*(parr+3)[2] =

3. (6 marks) Suppose you are working on a 32-bit machine where the size of an int is FOUR bytes, the size of a char is ONE byte and the size of pointers is FOUR bytes. See the following statements.

```
char *pa[] = {"12", "34", "56"};
int m[2][3] = {{1, 2, 3}, {4, 5, 6}};
int (*ppm)[2][3] = &m;
```

Give the values of the following expressions.

`sizeof(pa)=`

`sizeof(*pa)=`

`sizeof(**pa)=`

`sizeof(ppm)=`

`sizeof(*ppm)=`

`sizeof(**ppm)=`

4. (6 marks) Declare p.

p is a 5-element array of pointers to char.

p is a pointer to a 10-element char array.

p is a function that takes an int argument and returns a pointer to char.

p is a function that takes a char array and returns a pointer to int.

p is a pointer to a function that takes two int arguments and returns a pointer to an int.

p is a function that takes no arguments and returns a pointer to a function that takes an int argument and returns a pointer to a 10-element int array.

Programming Questions

5. (7 marks) Complete the following program that prompts user to enter two dates and then indicates which date comes earlier on the calendar.

```
#include <stdio.h>

/* Note: Program assumes years are in the same century. */

struct date {
    int month, day, year;
};

int compare_dates(struct date d1, struct date d2);
void put_date(struct date d);

int main(void)
{
    struct date d1, d2;

    printf("Enter first date (mm/dd/yy): ");
    /* for you to complete */
    printf("Enter second date (mm/dd/yy): ");
    /* for you to complete */

    if (compare_dates(d1, d2) < 0) {
        put_date(d1);
        printf(" is earlier than ");
        put_date(d2);
        printf("\n");
    } else {
        put_date(d2);
        printf(" is earlier than ");
        put_date(d1);
        printf("\n");
    }

    return 0;
}

int compare_dates(struct date d1, struct date d2)
{
    /* for you to complete */

    return 0;
}

void put_date(struct date d)
{
    printf("%d/%d/%.2d", d.month, d.day, d.year);
}
```

6. (5 marks) Rewrite the program in Question 5, using the following function prototype.

```
int compare_dates(struct date *, struct date *);
void put_date(struct date *);
```

7. (5 marks) Write function `swap_ptr` which swaps the values between `ptrp` and `ptrq`. You also need to provide the function prototype for `swap_ptr`.

```
int main(void)
{ int p = 11, q = 22;
  int *ptrp = &p, *ptrq = &q;
  int **ppp = &ptrp, **ppq = &ptrq;
  swap_ptr(ppp,ppq);          /* &ptrp, &ptrq passed */
                              /* to swap_ptr() */
  return 0;
}
```

8. (11 marks) Complete the following program. The program should provide user a text menu with 5 options (see below). Options 0~3 are implemented by functions. That is, making choice of these four options will call the corresponding functions. In this program, you need to use **arrays of pointers to functions** and call **these functions using pointers**.

Enter a choice:

- 0 Print the array of grades
- 1 Find the minimum grade
- 2 Find the maximum grade
- 3 Print the average on all tests for each student
- 4 End program

```
#include <stdio.h>
#define STUDENTS 3
#define EXAMS 4

/* function prototypes */
void minimum( int grades[][ EXAMS ], int pupils, int tests );
void maximum( int grades[][ EXAMS ], int pupils, int tests );
void average( int grades[][ EXAMS ], int pupils, int tests );
void printArray( int grades[][ EXAMS ], int pupils, int tests );
void printMenu( void );

int main(void)
{
    /* pointer to a function that takes as parameters a
       two-dimensional array and two integer values */
    void ( *processGrades[ 4 ] )( int [][] EXAMS ], int, int )
        = { printArray, minimum, maximum, average };

    int choice = 0; /* menu choice */

    /* array of student grades */
    int studentGrades[ STUDENTS ][ EXAMS ] = { { 77, 68, 86, 73 },
                                                { 96, 87, 89, 78 },
                                                { 70, 90, 86, 81 } };

    /* loop while user does not choose option 4 */
```

```

while ( choice != 4 ) {

    /* display menu and read user's choice */
    do {
        printMenu();
        scanf( "%d", &choice );
    } while ( choice < 0 || choice > 4 ); /* end do...while */

    /* pass choice into the array */
/* for you to complete */

    } /* end while */

    return 0; /* indicate successful termination */
} /* end main */

/* search for the minimum value */
void minimum( int grades[][ EXAMS ], int pupils, int tests )
{
    int i; /* loop counter */
    int j; /* loop counter */
    int lowGrade = 100; /* set lowGrade to highest possible score */

/* for you to complete */

    printf( "\n\tThe lowest grade is %d\n", lowGrade );
} /* end function minimum */

/* search for maximum value */
void maximum( int grades[][ EXAMS ], int pupils, int tests )
{
    int i; /* loop counter */
    int j; /* loop counter */
    int highGrade = 0; /* set highGrade to lowest possible score */

/* for you to complete */

    printf( "\n\tThe highest grade is %d\n", highGrade );
} /* end function maximum */

/* calculate average */
void average( int grades[][ EXAMS ], int pupils, int tests )
{
    int i; /* loop counter */
    int j; /* loop counter */
    int total; /* sum of all grades */

    printf( "\n" );

/* for you to complete */

} /* end function average */

/* print the contents of the array */
void printArray( int grades[][ EXAMS ], int pupils, int tests )
{
    int i; /* loop counter */
    int j; /* loop counter */

```

```

printf( "\n\t          [ 0 ] [ 1 ] [ 2 ] [ 3 ]" );

/* for you to complete */

printf( "\n" );
} /* end function printArray */

/* display the menu */
void printMenu( void )
{
    printf( "\n\tEnter a choice:\n"
           "\t 0  Print the array of grades\n"
           "\t 1  Find the minimum grade\n"
           "\t 2  Find the maximum grade\n"
           "\t 3  Print the average on all"
           " tests for each student\n"
           "\t 4  End program\n"
           "\t?  " );
} /* end function printMenu */

```

Submission Guidelines

- All Exercise Questions need to be put in a single text or pdf file with file name "eq".
- Each programming question should be a .c file. For example, the program for question 5 should be named as "q5.c". You need include the code provided with the questions so that the .c file can be compiled.
- You need provide appropriate comments to make your code readable. (Note that if your code is not working and there are no comments, you may loss all the marks.)
- We assume C89 standard. If you use C99 features, please highlight with comments.