

Name _____ KEY _____

Section _____

CS2303 A14
Systems Programming Concepts
Mid Term Exam
September 19, 2014

Question	Points	Score
0	1	
1	3	
2	5	
3	12	
4	16	
5	44	
Total	80	

Trivia Question (1 extra credit point)

0. (a) What is the capital of Azerbaijan?

Baku

-OR-

(b) Which country will host the 2018 World Cup?

Russia

(1 pt) 1a. Define **contiguous** memory in a computer.

Contiguous memory means that all the allocated memory is next to itself in one connected location in memory (virtual memory).

(2pts) b. Explain knowing how the C compiler allocates contiguous memory for a two-dimensional **array** is important. (Provide an example.)

Since C stores two-dimensional arrays by "row-order major", row elements are stored contiguous followed by the next row. This means you can point to the address of the first column of a row and then treat that row like a one-dimensional array. Example is an array of characters where each row is viewed as a string.

2. Given the following screen output from a ccc computer and assuming you are rek:

```
$ pwd
/rek/CS2303/q2
$ls -l
total 77
-r----- 1 rek cs2303a14 43 Sep 14 12:42 q1.c
-r-x----- 1 rek cs2303a14 188 Sep 14 12:45 q2.c
-rwx----- 1 rek rek      209 Sep 15 08:49 q.out
```

(1pts) a. Provide the command to create a new directory **midterm**.

mkdir midterm

(1pts) b. Provide the command that copies the file **q1.c** into **midterm**.

cp q1.c midterm/

(3 pts) Provide the command(s) that changes the access permissions on the file **q2.c** to:

read, write and execute for the owner
read and execute for the group
read and execute for other

chmod 755 q2.c

OR

chmod u+w q2.c

chmod g =rx, o=rx q2.c

(12 pts) 3. What is the output from this program? (You must show your work if you want partial credit)

```
#include <stdio.h>
#include <stdlib.h>
struct car {
    char *make;
    int row;
    int col;
};
typedef struct car Car;

void printCar(Car veh)
{
    printf ("C : %2d %2d %s\n", veh.row, veh.col, veh.make) ;
    return;
}

void printCarPtr(Car *veh)
{
    static cnt = 1;
    printf ("P%d: %-2d %-2d %s\n", cnt, veh->row, veh->col, veh->make);
    cnt++;
    return;
}

int main ()
{
    Car Toyo = {"Toyota", 12};
    Car Ford = {"Ford" , 7, 6};
    Car Hond = {"Honda" , 8, 5};
    Car *cPtr1, *cPtr2, *cPtr3;

    cPtr1 = malloc(sizeof(Car));
    cPtr2 = &Toyo;
    cPtr3 = &Hond;
    cPtr1->row = 4;
    cPtr1->col = cPtr2->col;

    cPtr1->make = Ford.make;
    cPtr2 = cPtr1;
    printCarPtr (cPtr2);

    cPtr3->col = Toyo.row;
    printCar (Hond);

    cPtr1->make = "Prius";
    cPtr1->row = cPtr3->col+2;
    cPtr1->col++;
    printCarPtr (cPtr1);

    free (cPtr1);
    printCarPtr (cPtr2);
    return;
}
```

Q3 OUTPUT Box

P1: 4 0 Ford
C : 8 12 Honda
P2: 14 1 Prius
P3: {all three ??}

- (16 pts.) 4. Given the partial program below, complete the code for the function **square**. The program reads in the length of the side of a square and a row and column. The function **square** creates a square with stars on the sides of the square inside the field array with the uppermost left corner of the square being at (row,col) inside the field. Assume the user input data keeps the square inside the field. Note - the **printfield** function is not shown.

```
#include <stdio.h>
#define SIZE 50
char BLANK = ' ';
char STAR = '*';

void square(char room[][SIZE], int r, int c, int s);
void printfield(char room[][SIZE]);

int main(){
int i, j, row, col, side;
char field [SIZE][SIZE];

for(i=0; i < SIZE; i++)
    for (j=0; j < SIZE; j++)
        field[i][j] = BLANK;

scanf ("%d %d %d", &side, &row, &col);
square (field, row, col, side);
printfield(field);
return 0;
}
```

```
void square(char room[SIZE][SIZE], int r, int c, int edge )
{

int i,j, r1, c1, e1;
r1 = r+1;
c1 = c;
i=1;
while (i<= edge)
{
    room[r][c] = STAR;
    room[r+edge-1][c] = STAR;
    c++;
    i++;
};
j=1;
e1 = edge-1;
while (j<=e1)
{
    room[r1][c1] = STAR;
    room[r1][c1+edge -1] = STAR;
    r1++;
    j++;
}
return;
}
```

(44 pts) 5. What is the output from this program? (You must show your work if you want partial credit)

```
#include <stdio.h>
#define LEN 5
#define DLEN (2*LEN)
int glob = 100;
int *gPtr = &glob;
void printr (const row[]){
    int i;
    printf("R :");
    for (i=0; i<LEN; i++)
        printf("%4d ", row[i]);
    printf("\n");
    return;
}
int zip (int k, float x){
    static int m = 4;
    (*gPtr)++;
    m = 1+m /3;
    x = x/3 - k;
    k = k % DLEN;
    printf("Z%d: %d %d %7.2f\n", m, *gPtr, k, x);
    return k;
}
float down (int f[LEN], int q, float y){
    int i;
    for (i=0; i<LEN; i++)
        f[i] = q-(int)y*i;
    glob++;
    printf("D :%6.2f\n", y);
    return 2*y;
}
int main(){
    int i,j,k = 3;
    float x = 24, y = 1.5, z;
    int ray[LEN] = {30, 205, 40};
    i = LEN-3;    j = y;
    z = ray[i-1]/k;
    k = k*glob;
    y = (float)k/ray[i];
    ray[i] = 54;
    glob = 2*glob;
    printf("M1: %5.2f %5.2f %5.2f\n", x,y,z);
    printf("M2: %3d %3d %3d %3d\n", i,j,k,glob);
    printr(ray);
    i = 3;    ray[3]= --i;
    printr(ray);
    z = 6.0;
    for (j=1; j<=2; j++){
        x = zip(ray[j],z);
        ray[j] = ray[LEN-(i--)];
        z = 2*z;
        printr(ray);
    }
    z = down(ray,10,z);
    printf("M3: %5.2f %5.2f %5.2f\n", x,y,z);
    printf("M4: %3d %3d %3d %3d\n", i,j,k,glob);
    printr(ray);
    return 0;
}
```

