Course: CSC 408
Final Exam date: 10th December 2024
Name: Khadichabonu Valieva (w10118633)
Total time: 10.45 am- 1.15 pm (150 min)
Total marks: 130

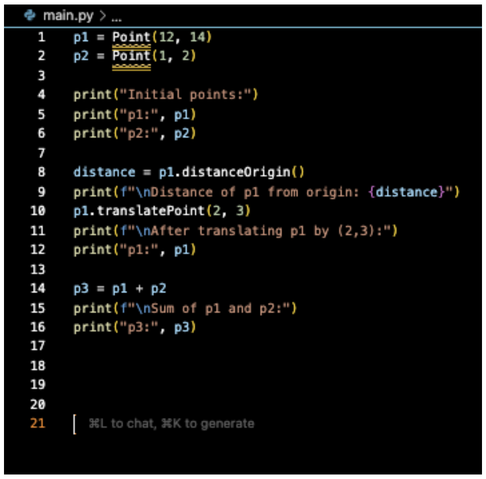
Questions	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Total	Percent
Marks	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	130	%
Marks obtained																	

Directions: Write any 13 questions out of 15 in the best possible syntax. Do not write the whole code, only write brief code as the question demands. You will get extra credit if you answer all questions. Do not write any code from the internet as the answer sheet will pass through AI generated code detection software. Also, no group work is allowed as any similarity of code will result in F grade.

1. Create a python class of cartesian coordinate Point and write a constructor, a translatePoint member function (which translate the point object x coordinate to deltaX and y coordinate to deltaY), a distanceOrigin member function(which finds the distance of the point object from origin (0,0)), an overloaded print function (which prints the x coordinate and y coordinate of the point), an overloaded add function which outputs a new point (formed by adding the coordinates of two point objects). Implement each member function in the code. (Marks 10)

```python
import math
class Point:
    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y
    def translate_point(self, delta_x, delta_y):
        self.x += delta_x
        self.y += delta_y
    def distance_from_origin(self):
        return math.sqrt(self.x**2 + self.y**2)
    def __str__(self):
        return f"Point({self.x}, {self.y})"
    def __add__(self, other):
        return Point(self.x + other.x, self.y + other.y)
```

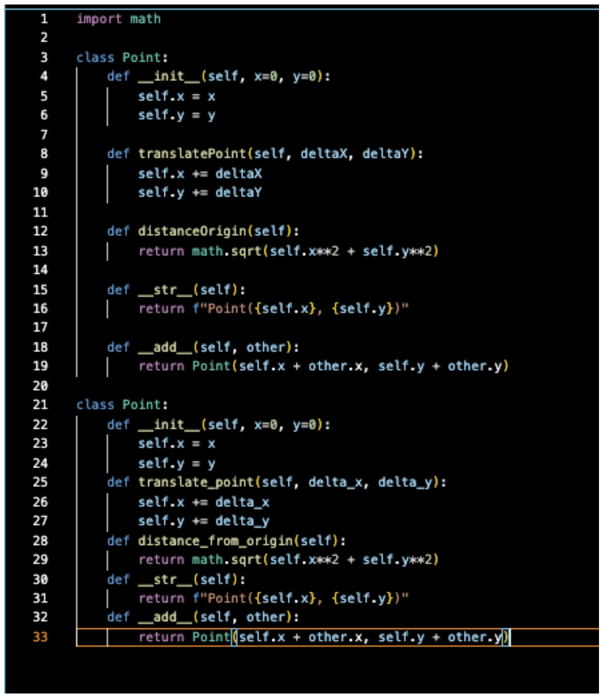
Screenshot:



main.py > ...

```python
p1 = Point(12, 14)
p2 = Point(1, 2)

print("Initial points:")
print("p1:", p1)
print("p2:", p2)

distance = p1.distanceOrigin()
print(f"\nDistance of p1 from origin: {distance}")
p1.translatePoint(2, 3)
print(f"\nAfter translating p1 by (2,3):")
print("p1:", p1)

p3 = p1 + p2
print(f"\nSum of p1 and p2:")
print("p3:", p3)




```

⌘L to chat, ⌘K to generate



```python
import math

class Point:
    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y

    def translatePoint(self, deltaX, deltaY):
        self.x += deltaX
        self.y += deltaY

    def distanceOrigin(self):
        return math.sqrt(self.x**2 + self.y**2)

    def __str__(self):
        return f"Point({self.x}, {self.y})"

    def __add__(self, other):
        return Point(self.x + other.x, self.y + other.y)

class Point:
    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y
    def translate_point(self, delta_x, delta_y):
        self.x += delta_x
        self.y += delta_y
    def distance_from_origin(self):
        return math.sqrt(self.x**2 + self.y**2)
    def __str__(self):
        return f"Point({self.x}, {self.y})"
    def __add__(self, other):
        return Point(self.x + other.x, self.y + other.y)
```

2. int i = 100189;
short s = i;

cout << s << endl;

Explain the output by showing the bit pattern of the integer assignment of i and type casting to s (Marks 10).

Here integer i is being converted to short. To explain the process:

1. Integer I=100189, in binary (32bit) is 00000000000000011000011101011101
2. Short is typically 16 bits, so when we assigned int to short, we only take the rightmost 16 bits 1000011101011101
3. Since the leftmost big of the short (1) is set, this translates into being a negative number in two's complement.
4. To get the actual value of that 16 bit, we invert all bits and add 1, and output as a negative number: 0111100010100010 + 1 = 0111100010100011
5. This gives us -31555 in decimal

(To convert decimal to binary and binary to decimal I used the online converter: https://www.rapidtables.com/convert/number/decimal-to-binary.html?x=100189)

3. In the following struct

```
struct student {

   char name[20];
   char id[10];
   int testScore[3];
   float avgScore;
   char grade;

}s;
```

If the address of s is 00FAFD50 then what is the starting address of name, id, textScore, avgScore and grade? (Marks 10)

Base address is 00FAFD50. Assuming that we take standard alignment rules, we get the following. Also, I used decimals in the solution to show the steps, however when I actually add, I first convert them to hex and output the hex result:

name [20] – starts at base address, so it's base address => 00FAFD50

id [10] – starts right after name so it's base + 20 = 00FAFD50+20 => 00FAFD64

testScore [30] – starts after id, needs 4 bytes alignment, takes 12 bytes (3 integers, 4 bytes each), so address is aligned to 4-byte boundary => 00FAFD70

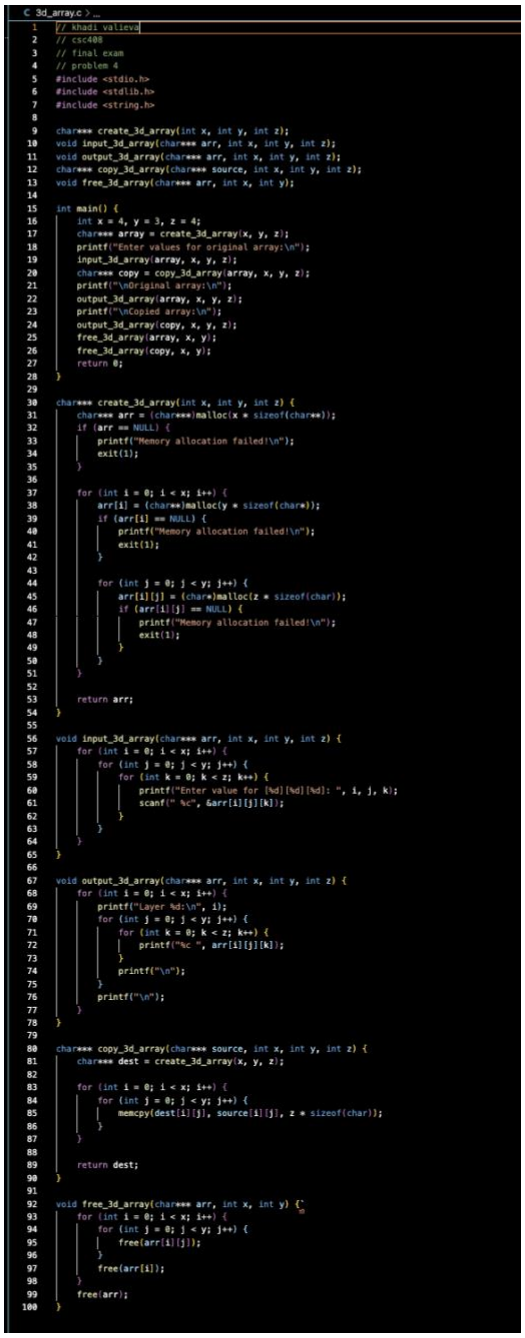
avgScore – also needs 4 byte alignment, starts after testScore, base + 20+10+12 => 00FAFD7C

grade – it's a single char, starts after avgScore, base + 20 + 10 + 12 + 4 => 00FAFD80

4. Write C code to dynamically create a 4X3X4 char array using a triple pointer. Write functions to input and output values in the array and copy the content of the array to another array using memcpy. Also delete the allocation from the heap. (Marks 10)

```c
char*** create_3d_array(int x, int y, int z);
void input_3d_array(char*** arr, int x, int y, int z);
void output_3d_array(char*** arr, int x, int y, int z);
char*** copy_3d_array(char*** source, int x, int y, int z);
void free_3d_array(char*** arr, int x, int y);

int main() {
    int x = 4, y = 3, z = 4;
    char*** array = create_3d_array(x, y, z);
    printf("Enter values for original array:\n");
    input_3d_array(array, x, y, z);
    char*** copy = copy_3d_array(array, x, y, z);
    printf("\nOriginal array:\n");
    output_3d_array(array, x, y, z);
    printf("\nCopied array:\n");
    output_3d_array(copy, x, y, z);
    free_3d_array(array, x, y);
    free_3d_array(copy, x, y);
    return 0;
}

char*** create_3d_array(int x, int y, int z) {
    char*** arr = (char***)malloc(x * sizeof(char**));
    if (arr == NULL) {
        printf("Memory allocation failed!\n");
        exit(1);
    }

    for (int i = 0; i < x; i++) {
        arr[i] = (char**)malloc(y * sizeof(char*));
        if (arr[i] == NULL) {
            printf("Memory allocation failed!\n");
            exit(1);
        }

        for (int j = 0; j < y; j++) {
            arr[i][j] = (char*)malloc(z * sizeof(char));
            if (arr[i][j] == NULL) {
                printf("Memory allocation failed!\n");
                exit(1);
            }
        }
    }

    return arr;
}

void input_3d_array(char*** arr, int x, int y, int z) {
    for (int i = 0; i < x; i++) {
        for (int j = 0; j < y; j++) {
            for (int k = 0; k < z; k++) {
                printf("Enter value for [%d][%d][%d]: ", i, j, k);
                scanf(" %c", &arr[i][j][k]);
            }
        }
    }
}

void output_3d_array(char*** arr, int x, int y, int z) {
    for (int i = 0; i < x; i++) {
        printf("Layer %d:\n", i);
        for (int j = 0; j < y; j++) {
            for (int k = 0; k < z; k++) {
                printf("%c ", arr[i][j][k]);
            }
            printf("\n");
        }
        printf("\n");
    }
}

char*** copy_3d_array(char*** source, int x, int y, int z) {
    char*** dest = create_3d_array(x, y, z);

    for (int i = 0; i < x; i++) {
        for (int j = 0; j < y; j++) {
            memcpy(dest[i][j], source[i][j], z * sizeof(char));
        }
    }

    return dest;
}

void free_3d_array(char*** arr, int x, int y) {`
    for (int i = 0; i < x; i++) {
        for (int j = 0; j < y; j++) {
            free(arr[i][j]);
        }
        free(arr[i]);
    }
    free(arr);}
```

Screenshot:



3d_array.c > ...

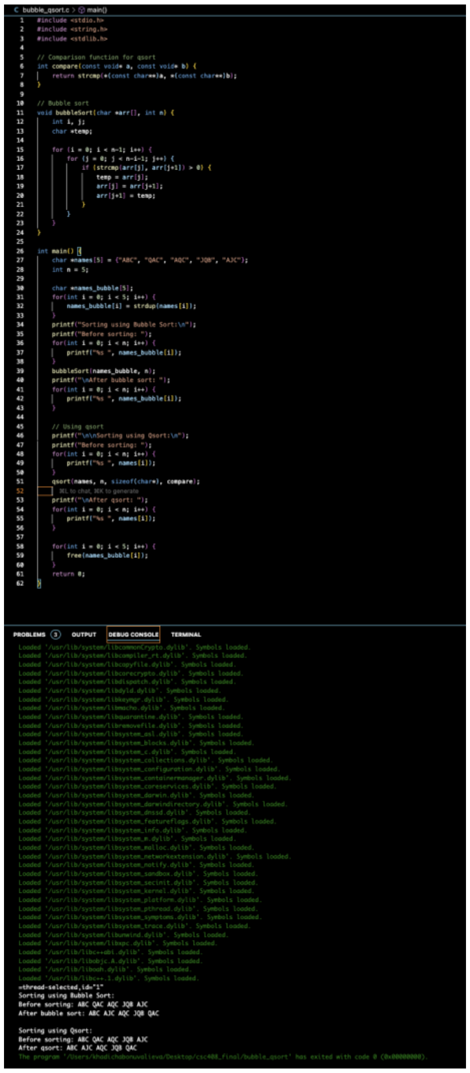
```c
// khadi valieva
// csc408
// final exam
// problem 4
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

char*** create_3d_array(int x, int y, int z);
void input_3d_array(char*** arr, int x, int y, int z);
void output_3d_array(char*** arr, int x, int y, int z);
char*** copy_3d_array(char*** source, int x, int y, int z);
void free_3d_array(char*** arr, int x, int y);

int main() {
    int x = 4, y = 3, z = 4;
    char*** array = create_3d_array(x, y, z);
    printf("Enter values for original array:\n");
    input_3d_array(array, x, y, z);
    char*** copy = copy_3d_array(array, x, y, z);
    printf("\nOriginal array:\n");
    output_3d_array(array, x, y, z);
    printf("\nCopied array:\n");
    output_3d_array(copy, x, y, z);
    free_3d_array(array, x, y);
    free_3d_array(copy, x, y);
    return 0;
}

char*** create_3d_array(int x, int y, int z) {
    char*** arr = (char***)malloc(x * sizeof(char**));
    if (arr == NULL) {
        printf("Memory allocation failed!\n");
        exit(1);
    }

    for (int i = 0; i < x; i++) {
        arr[i] = (char**)malloc(y * sizeof(char*));
        if (arr[i] == NULL) {
            printf("Memory allocation failed!\n");
            exit(1);
        }

        for (int j = 0; j < y; j++) {
            arr[i][j] = (char*)malloc(z * sizeof(char));
            if (arr[i][j] == NULL) {
                printf("Memory allocation failed!\n");
                exit(1);
            }
        }
    }

    return arr;
}

void input_3d_array(char*** arr, int x, int y, int z) {
    for (int i = 0; i < x; i++) {
        for (int j = 0; j < y; j++) {
            for (int k = 0; k < z; k++) {
                printf("Enter value for [%d][%d][%d]: ", i, j, k);
                scanf(" %c", &arr[i][j][k]);
            }
        }
    }
}

void output_3d_array(char*** arr, int x, int y, int z) {
    for (int i = 0; i < x; i++) {
        printf("Layer %d:\n", i);
        for (int j = 0; j < y; j++) {
            for (int k = 0; k < z; k++) {
                printf("%c ", arr[i][j][k]);
            }
            printf("\n");
        }
        printf("\n");
    }
}

char*** copy_3d_array(char*** source, int x, int y, int z) {
    char*** dest = create_3d_array(x, y, z);

    for (int i = 0; i < x; i++) {
        for (int j = 0; j < y; j++) {
            memcpy(dest[i][j], source[i][j], z * sizeof(char));
        }
    }

    return dest;
}

void free_3d_array(char*** arr, int x, int y) {
    for (int i = 0; i < x; i++) {
        for (int j = 0; j < y; j++) {
            free(arr[i][j]);
        }
        free(arr[i]);
    }
    free(arr);
}
```

5. Use library generic bubblesort and qsort in C to sort the following c-string (Marks 10)

```c
char *names[5] = { "ABC", "QAC", "AQC", "JQB", "AJC" };
```

```c
// Comparison function for qsort
int compare(const void* a, const void* b) {
    return strcmp(*(const char**)a, *(const char**)b);
}

// Bubble sort
void bubbleSort(char *arr[], int n) {
    int i, j;
    char *temp;

    for (i = 0; i < n-1; i++) {
        for (j = 0; j < n-i-1; j++) {
            if (strcmp(arr[j], arr[j+1]) > 0) {
                temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}

int main() {
    char *names[5] = {"ABC", "QAC", "AQC", "JQB", "AJC"};
    int n = 5;

    char *names_bubble[5];
    for(int i = 0; i < 5; i++) {
        names_bubble[i] = strdup(names[i]);
    }
    printf("Sorting using Bubble Sort:\n");
    printf("Before sorting: ");
    for(int i = 0; i < n; i++) {
        printf("%s ", names_bubble[i]);
    }
    bubbleSort(names_bubble, n);
    printf("\nAfter bubble sort: ");
    for(int i = 0; i < n; i++) {
        printf("%s ", names_bubble[i]);
    }

    // Using qsort
    printf("\n\nSorting using Qsort:\n");
    printf("Before sorting: ");
    for(int i = 0; i < n; i++) {
        printf("%s ", names[i]);
    }
    qsort(names, n, sizeof(char*), compare);

    printf("\nAfter qsort: ");
    for(int i = 0; i < n; i++) {
        printf("%s ", names[i]);
    }

    for(int i = 0; i < 5; i++) {
        free(names_bubble[i]);
    }
    return 0;
}
```

Screenshot:

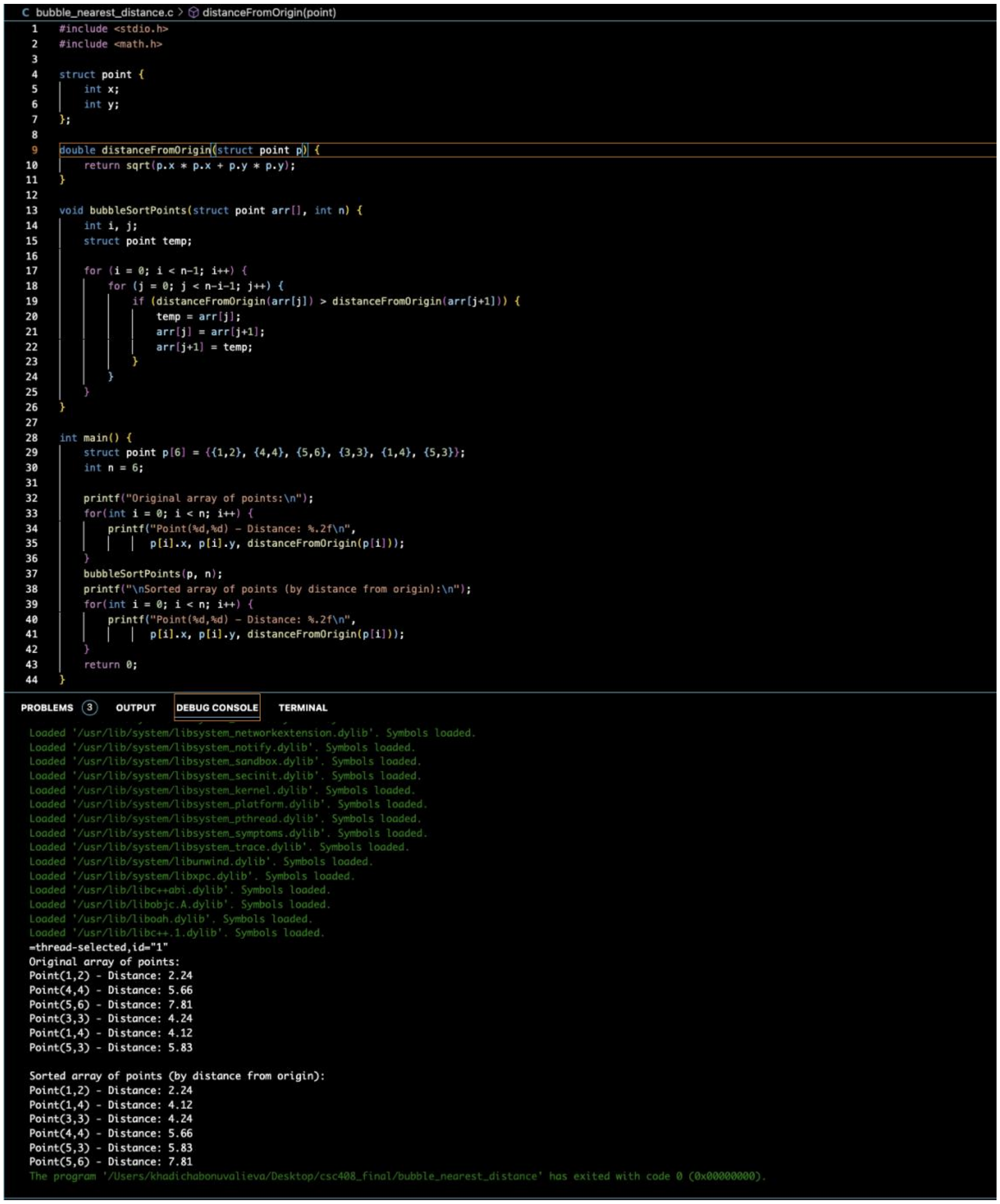


```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

// Comparison function for qsort
int compare(const void* a, const void* b) {
    return strcmp(*(const char**)a, *(const char**)b);
}

// Bubble sort
void bubbleSort(char *arr[], int n) {
    int i, j;
    char *temp;

    for (i = 0; i < n-1; i++) {
        for (j = 0; j < n-i-1; j++) {
            if (strcmp(arr[j], arr[j+1]) > 0) {
                temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}

int main() {
    char *names[5] = {"ABC", "QAC", "AQC", "JQB", "AJC"};
    int n = 5;

    char *names_bubble[5];
    for(int i = 0; i < 5; i++) {
        names_bubble[i] = strdup(names[i]);
    }
    printf("Sorting using Bubble Sort:\n");
    printf("Before sorting: ");
    for(int i = 0; i < n; i++) {
        printf("%s ", names_bubble[i]);
    }
    bubbleSort(names_bubble, n);
    printf("\nAfter bubble sort: ");
    for(int i = 0; i < n; i++) {
        printf("%s ", names_bubble[i]);
    }

    // Using qsort
    printf("\n\nSorting using Qsort:\n");
    printf("Before sorting: ");
    for(int i = 0; i < n; i++) {
        printf("%s ", names[i]);
    }
    qsort(names, n, sizeof(char*), compare);

    printf("\nAfter qsort: ");
    for(int i = 0; i < n; i++) {
        printf("%s ", names[i]);
    }

    for(int i = 0; i < 5; i++) {
        free(names_bubble[i]);
    }
    return 0;
}
```

```
Loaded '/usr/lib/system/libcommonCrypto.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libcompiler_rt.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libcopyfile.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libcorecrypto.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libdispatch.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libdyld.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libkeymgr.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libmacho.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libquarantine.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libremovefile.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_asl.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_blocks.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_c.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_collections.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_configuration.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_containermanager.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_coreservices.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_darwin.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_darwindirectory.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_dnssd.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_featureflags.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_info.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_m.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_malloc.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_networkextension.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_notify.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_sandbox.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_secinit.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_kernel.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_platform.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_pthread.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_symptoms.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_trace.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libunwind.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libxpc.dylib'. Symbols loaded.
Loaded '/usr/lib/libc++abi.dylib'. Symbols loaded.
Loaded '/usr/lib/libobjc.A.dylib'. Symbols loaded.
Loaded '/usr/lib/liboah.dylib'. Symbols loaded.
Loaded '/usr/lib/libc++.1.dylib'. Symbols loaded.
=thread-selected,id="1"
Sorting using Bubble Sort:
Before sorting: ABC QAC AQC JQB AJC
After bubble sort: ABC AJC AQC JQB QAC

Sorting using Qsort:
Before sorting: ABC QAC AQC JQB AJC
After qsort: ABC AJC AQC JQB QAC
The program '/Users/khadichabonuvalieva/Desktop/csc408_final/bubble_qsort' has exited with code 0 (0x00000000).
```

6. For 6 point objects struct point p[6] ={{1,2}, {4,4}, {5,6}, {3,3}, {1,4}, {5,3}}. Write a generic bubblesort in C code to sort the points based on their nearest distance to the center(0,0). (Marks 10)

```c
struct point {

   int x;

   int y;

};


double distanceFromOrigin(struct point p) {

   return sqrt(p.x * p.x + p.y * p.y);

}


void bubbleSortPoints(struct point arr[], int n) {
   int i, j;
   struct point temp;

   for (i = 0; i < n-1; i++) {
       for (j = 0; j < n-i-1; j++) {
           if (distanceFromOrigin(arr[j]) > distanceFromOrigin(arr[j+1])) {
               temp = arr[j];
               arr[j] = arr[j+1];
               arr[j+1] = temp;
           }
       }
   }
}

int main() {
   struct point p[6] = {{1,2}, {4,4}, {5,6}, {3,3}, {1,4}, {5,3}};
   int n = 6;

   printf("Original array of points:\n");
   for(int i = 0; i < n; i++) {
       printf("Point(%d,%d) - Distance: %.2f\n",
             p[i].x, p[i].y, distanceFromOrigin(p[i]));
   }
   bubbleSortPoints(p, n);
   printf("\nSorted array of points (by distance from origin):\n");
   for(int i = 0; i < n; i++) {
       printf("Point(%d,%d) - Distance: %.2f\n",
             p[i].x, p[i].y, distanceFromOrigin(p[i]));
   }
   return 0;
}
```



```c
#include <stdio.h>
#include <math.h>

struct point {
    int x;
    int y;
};

double distanceFromOrigin(struct point p) {
    return sqrt(p.x * p.x + p.y * p.y);
}

void bubbleSortPoints(struct point arr[], int n) {
    int i, j;
    struct point temp;

    for (i = 0; i < n-1; i++) {
        for (j = 0; j < n-i-1; j++) {
            if (distanceFromOrigin(arr[j]) > distanceFromOrigin(arr[j+1])) {
                temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}

int main() {
    struct point p[6] = {{1,2}, {4,4}, {5,6}, {3,3}, {1,4}, {5,3}};
    int n = 6;

    printf("Original array of points:\n");
    for(int i = 0; i < n; i++) {
        printf("Point(%d,%d) - Distance: %.2f\n",
               p[i].x, p[i].y, distanceFromOrigin(p[i]));
    }
    bubbleSortPoints(p, n);
    printf("\nSorted array of points (by distance from origin):\n");
    for(int i = 0; i < n; i++) {
        printf("Point(%d,%d) - Distance: %.2f\n",
               p[i].x, p[i].y, distanceFromOrigin(p[i]));
    }
    return 0;
}
```

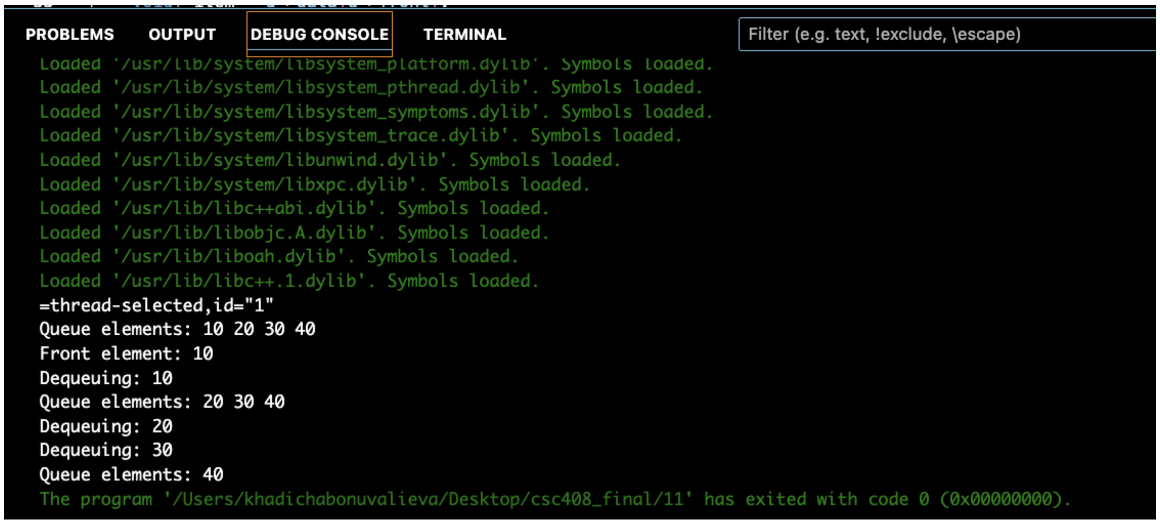
```
Loaded '/usr/lib/system/libsystem_networkextension.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_notify.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_sandbox.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_secinit.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_kernel.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_platform.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_pthread.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_symptoms.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_trace.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libunwind.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libxpc.dylib'. Symbols loaded.
Loaded '/usr/lib/libc++abi.dylib'. Symbols loaded.
Loaded '/usr/lib/libobjc.A.dylib'. Symbols loaded.
Loaded '/usr/lib/liboah.dylib'. Symbols loaded.
Loaded '/usr/lib/libc++.1.dylib'. Symbols loaded.
=thread-selected,id="1"
Original array of points:
Point(1,2) - Distance: 2.24
Point(4,4) - Distance: 5.66
Point(5,6) - Distance: 7.81
Point(3,3) - Distance: 4.24
Point(1,4) - Distance: 4.12
Point(5,3) - Distance: 5.83

Sorted array of points (by distance from origin):
Point(1,2) - Distance: 2.24
Point(1,4) - Distance: 4.12
Point(3,3) - Distance: 4.24
Point(4,4) - Distance: 5.66
Point(5,3) - Distance: 5.83
Point(5,6) - Distance: 7.81
The program '/Users/khadichabonuvalieva/Desktop/csc408_final/bubble_nearest_distance' has exited with code 0 (0x00000000).
```

7. Implement the enqueue, dequeue, and front functions of generic array implementation of a queue in C. (Marks 10)

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 10

typedef struct {
    void* data[MAX];
    int front, rear, size;
} Queue;
void initQueue(Queue* q) {
    q->front = 0;
    q->rear = -1;
    q->size = 0;
}
int enqueue(Queue* q, void* item) {
    if (q->size == MAX) {
        printf("Queue is full!\n");
        return -1;
    }
    q->rear = (q->rear + 1) % MAX;
    q->data[q->rear] = item;
    q->size++;
    return 0;
}
void* dequeue(Queue* q) {
    if (q->size == 0) {
        printf("Queue is empty!\n");
        return NULL;
    }
    void* item = q->data[q->front];
    q->front = (q->front + 1) % MAX;
    q->size--;
    return item;
}
void* front(Queue* q) {
```

```c
    if (q->size == 0) {
        printf("Queue is empty!\n");
        return NULL;
    }
    return q->data[q->front];
}
void displayQueue(Queue* q) {
    if (q->size == 0) {
        printf("Queue is empty!\n");
        return;
    }
    int i = q->front;
    printf("Queue elements: ");
    for (int j = 0; j < q->size; j++) {
        printf("%d ", *(int*)q->data[i]);
        i = (i + 1) % MAX;
    }
    printf("\n");
}
int main() {
    Queue q;
    initQueue(&q);
    int a = 10, b = 20, c = 30, d = 40;
    enqueue(&q, &a);
    enqueue(&q, &b);
    enqueue(&q, &c);
    enqueue(&q, &d);
    displayQueue(&q);
    printf("Front element: %d\n", *(int*)front(&q));
    printf("Dequeuing: %d\n", *(int*)dequeue(&q));
    displayQueue(&q);
    printf("Dequeuing: %d\n", *(int*)dequeue(&q));
    printf("Dequeuing: %d\n", *(int*)dequeue(&q));
    displayQueue(&q);
    return 0;
}
```

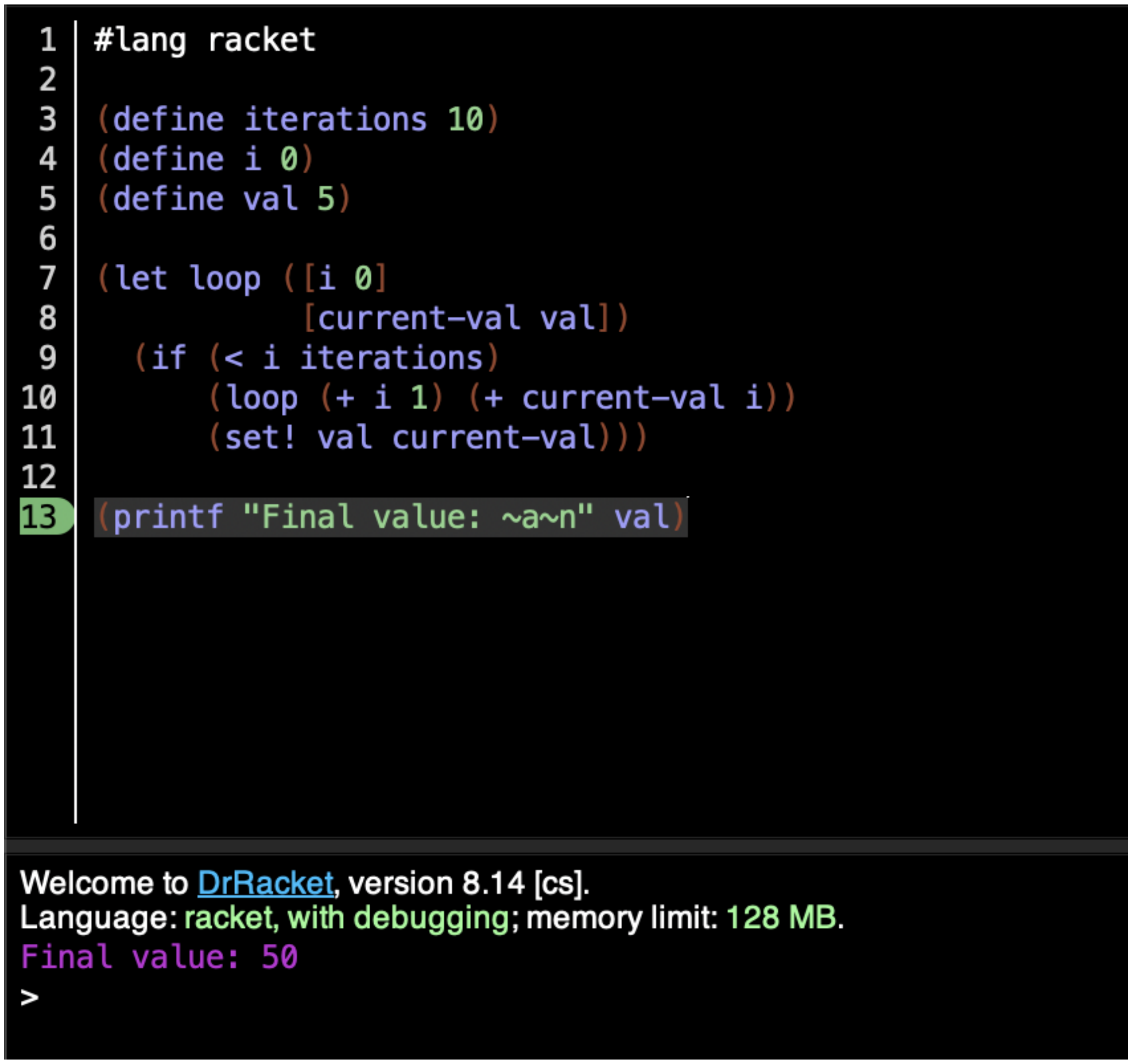
Results:



```
Loaded '/usr/lib/system/libsystem_platform.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_pthread.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_symptoms.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libsystem_trace.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libunwind.dylib'. Symbols loaded.
Loaded '/usr/lib/system/libxpc.dylib'. Symbols loaded.
Loaded '/usr/lib/libc++abi.dylib'. Symbols loaded.
Loaded '/usr/lib/libobjc.A.dylib'. Symbols loaded.
Loaded '/usr/lib/liboah.dylib'. Symbols loaded.
Loaded '/usr/lib/libc++.1.dylib'. Symbols loaded.
=thread-selected,id="1"
Queue elements: 10 20 30 40
Front element: 10
Dequeuing: 10
Queue elements: 20 30 40
Dequeuing: 20
Dequeuing: 30
Queue elements: 40
The program '/Users/khadichabonuvalieva/Desktop/csc408_final/11' has exited with code 0 (0x00000000).
```

8. Convert the following code into an assembly code (Marks 10)

```
int iterations = 10;
int i = 0;
int val = 5;
for (i = 0; i < iterations; i++)
        val += i;
```



```racket
#lang racket

(define iterations 10)
(define i 0)
(define val 5)

(let loop ([i 0]
           [current-val val])
  (if (< i iterations)
      (loop (+ i 1) (+ current-val i))
      (set! val current-val)))

(printf "Final value: ~a~n" val)
```

```
Welcome to DrRacket, version 8.14 [cs].
Language: racket, with debugging; memory limit: 128 MB.
Final value: 50
>
```

9. Draw an activation record of the function call A() showing the position of the stack pointer. Mention the total size in bytes of the activation record of A(). (Marks 10)

```c
void A() {
        int x;
        short b[4];
        double c;
        B();
        C();
}
```

```
void B() {
        int m;
        char* n;
        char* r[2];
        C();
}

void C() {
        double j[5];
        int k;
}
```

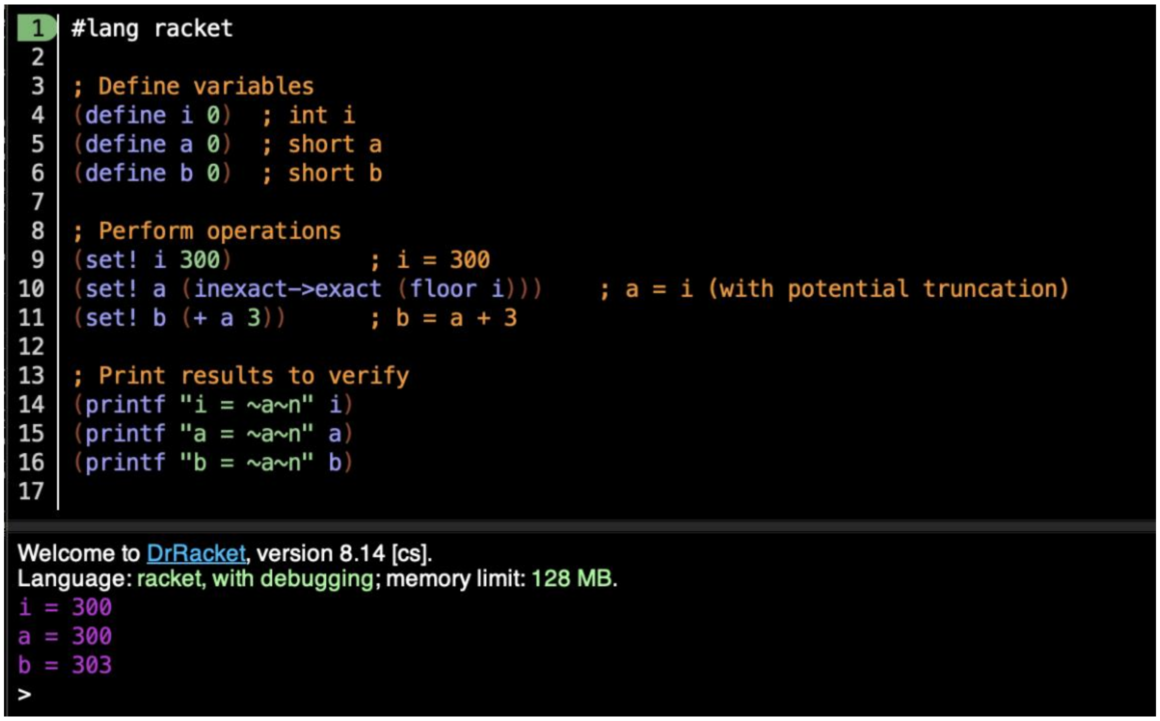
Return address + old EBP	4+4 = 8 bytes	<- old EBP
x	4 bytes (int)	<- new EBP
b[0] b[1] b[2] b[3]	2 bytes (short) * 4 = 8 bytes total	
c	8 bytes (double)	
		<- ESP

Return address + old EBP = 8
x = 4
b[4] = 8
c = 8

Total: 8+4+8+8 = 28 bytes (I didn't allocate any space for memory, since B() and C() don't return any parameters.

10. Write the assembly code of the following (Marks 10)

```
int i;
short a;
short b;
i = 300;
a = i;
b = a + 3;
```

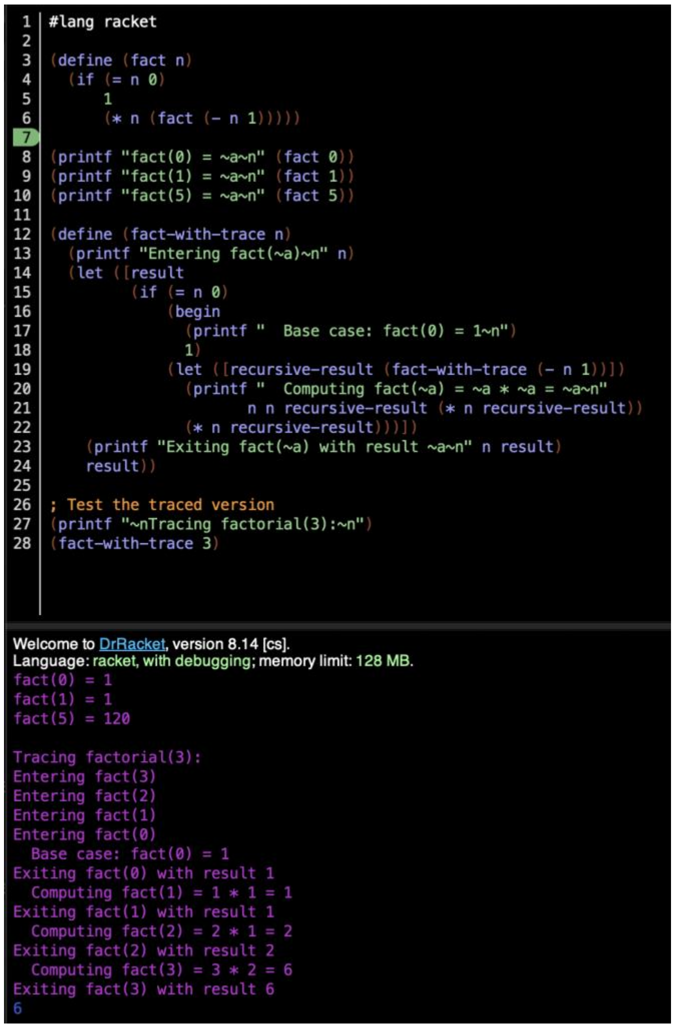


```racket
#lang racket

; Define variables
(define i 0)  ; int i
(define a 0)  ; short a
(define b 0)  ; short b

; Perform operations
(set! i 300)            ; i = 300
(set! a (inexact->exact (floor i)))    ; a = i (with potential truncation)
(set! b (+ a 3))        ; b = a + 3

; Print results to verify
(printf "i = ~a~n" i)
(printf "a = ~a~n" a)
(printf "b = ~a~n" b)
```

```
Welcome to DrRacket, version 8.14 [cs].
Language: racket, with debugging; memory limit: 128 MB.
i = 300
a = 300
b = 303
>
```

11. Write the assembly code of the following function fact. Also draw the activation record of the function call (Marks 10)

```c
int fact(int n) {

        if (n == 0)
                return 1;
        return n * fact(n - 1);
}
```



```racket
#lang racket

(define (fact n)
  (if (= n 0)
      1
      (* n (fact (- n 1)))))

(printf "fact(0) = ~a~n" (fact 0))
(printf "fact(1) = ~a~n" (fact 1))
(printf "fact(5) = ~a~n" (fact 5))

(define (fact-with-trace n)
  (printf "Entering fact(~a)~n" n)
  (let ([result
         (if (= n 0)
             (begin
               (printf "  Base case: fact(0) = 1~n")
               1)
             (let ([recursive-result (fact-with-trace (- n 1))])
               (printf "  Computing fact(~a) = ~a * ~a = ~a~n"
                       n n recursive-result (* n recursive-result))
               (* n recursive-result)))])
    (printf "Exiting fact(~a) with result ~a~n" n result)
    result))

; Test the traced version
(printf "~nTracing factorial(3):~n")
(fact-with-trace 3)
```

```
Welcome to DrRacket, version 8.14 [cs].
Language: racket, with debugging; memory limit: 128 MB.
fact(0) = 1
fact(1) = 1
fact(5) = 120

Tracing factorial(3):
Entering fact(3)
Entering fact(2)
Entering fact(1)
Entering fact(0)
  Base case: fact(0) = 1
Exiting fact(0) with result 1
  Computing fact(1) = 1 * 1 = 1
Exiting fact(1) with result 1
  Computing fact(2) = 2 * 1 = 2
Exiting fact(2) with result 2
  Computing fact(3) = 3 * 2 = 6
Exiting fact(3) with result 6
6
```

12. There are 10 agents to sell tickets and totalTickets= 400. Use multithreading representing the agents each trying to sell the total tickets without any race condition using semaphore on the following function. (Marks 10)

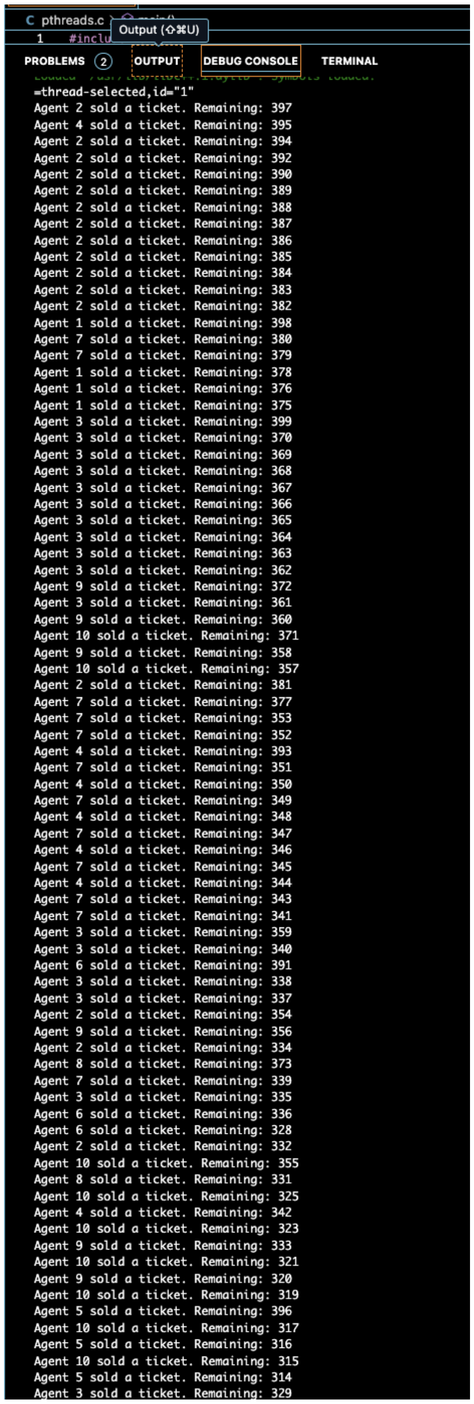
```c
void sellTickets(int agent, int* totalTickets) {
…….

}
```

Use print function when agents sell the ticket and when all tickets are sold.

```c
#include <stdio.h>
```

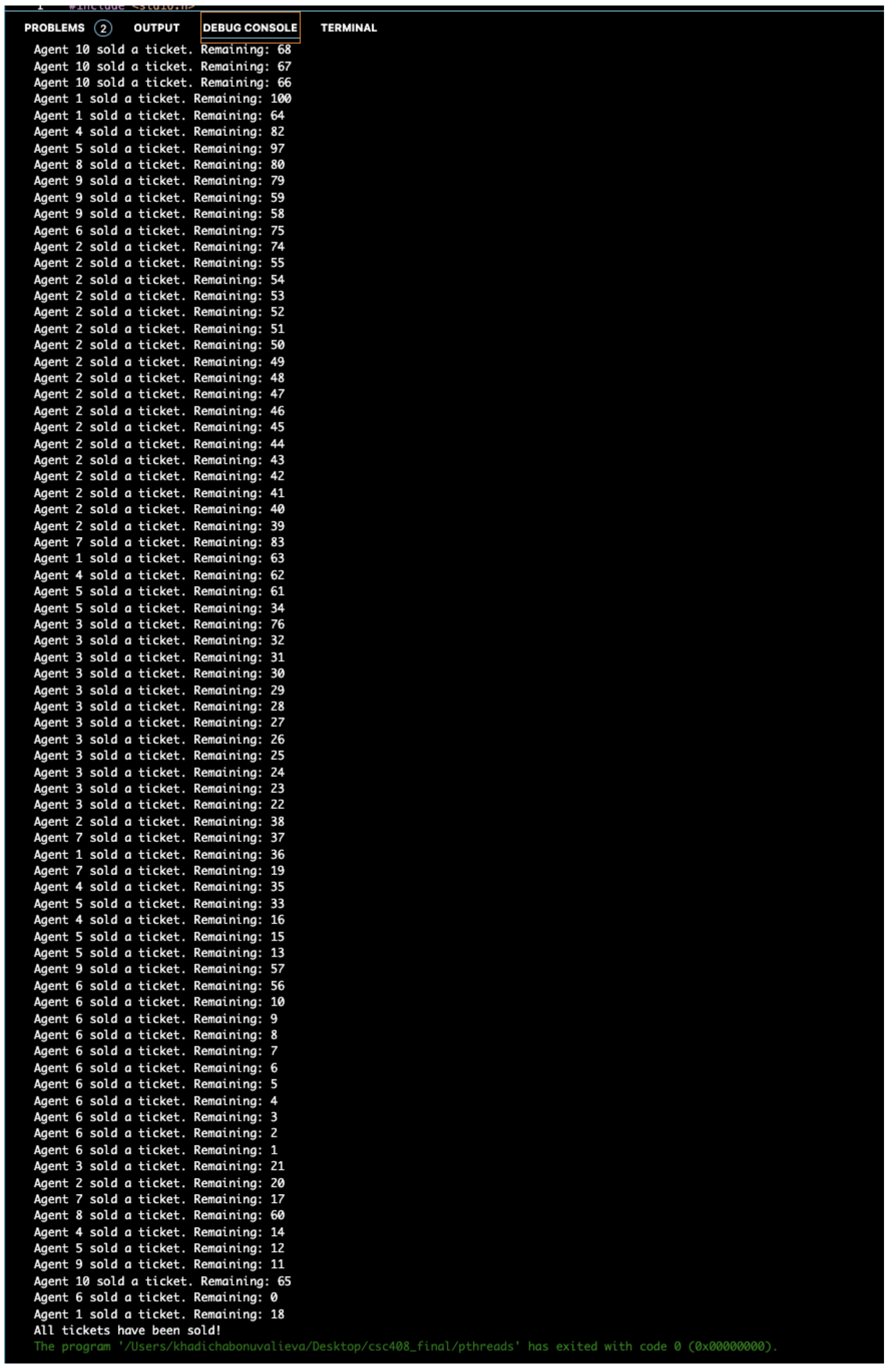
```c
#include <pthread.h>
#include <semaphore.h>

#define TOTAL_TICKETS 400
#define AGENTS 10
int totalTickets = TOTAL_TICKETS;
sem_t semaphore;

void* sellTickets(void* arg) {
    int agent = *((int*)arg);
    while (1) {
        sem_wait(&semaphore);
        if (totalTickets > 0) {
            totalTickets--;
            printf("Agent %d sold a ticket. Remaining: %d\n", agent, totalTickets);
        } else {
            sem_post(&semaphore);
            break;
        }
        sem_post(&semaphore);
    }
    return NULL;
}

int main() {
    pthread_t threads[AGENTS];
    int agentIds[AGENTS];
    sem_init(&semaphore, 0, 1);
    for (int i = 0; i < AGENTS; i++) {
        agentIds[i] = i + 1;
        pthread_create(&threads[i], NULL, sellTickets, &agentIds[i]);}
    for (int i = 0; i < AGENTS; i++) {
        pthread_join(threads[i], NULL);
    }
    printf("All tickets have been sold!\n");
    sem_destroy(&semaphore);
    return 0;
}
```

Resutls:



```
=thread-selected,id="1"
Agent 2 sold a ticket. Remaining: 397
Agent 4 sold a ticket. Remaining: 395
Agent 2 sold a ticket. Remaining: 394
Agent 2 sold a ticket. Remaining: 392
Agent 2 sold a ticket. Remaining: 390
Agent 2 sold a ticket. Remaining: 389
Agent 2 sold a ticket. Remaining: 388
Agent 2 sold a ticket. Remaining: 387
Agent 2 sold a ticket. Remaining: 386
Agent 2 sold a ticket. Remaining: 385
Agent 2 sold a ticket. Remaining: 384
Agent 2 sold a ticket. Remaining: 383
Agent 2 sold a ticket. Remaining: 382
Agent 1 sold a ticket. Remaining: 398
Agent 7 sold a ticket. Remaining: 380
Agent 7 sold a ticket. Remaining: 379
Agent 1 sold a ticket. Remaining: 378
Agent 1 sold a ticket. Remaining: 376
Agent 1 sold a ticket. Remaining: 375
Agent 3 sold a ticket. Remaining: 399
Agent 3 sold a ticket. Remaining: 370
Agent 3 sold a ticket. Remaining: 369
Agent 3 sold a ticket. Remaining: 368
Agent 3 sold a ticket. Remaining: 367
Agent 3 sold a ticket. Remaining: 366
Agent 3 sold a ticket. Remaining: 365
Agent 3 sold a ticket. Remaining: 364
Agent 3 sold a ticket. Remaining: 363
Agent 3 sold a ticket. Remaining: 362
Agent 9 sold a ticket. Remaining: 372
Agent 3 sold a ticket. Remaining: 361
Agent 9 sold a ticket. Remaining: 360
Agent 10 sold a ticket. Remaining: 371
Agent 9 sold a ticket. Remaining: 358
Agent 10 sold a ticket. Remaining: 357
Agent 2 sold a ticket. Remaining: 381
Agent 7 sold a ticket. Remaining: 377
Agent 7 sold a ticket. Remaining: 353
Agent 7 sold a ticket. Remaining: 352
Agent 4 sold a ticket. Remaining: 393
Agent 7 sold a ticket. Remaining: 351
Agent 4 sold a ticket. Remaining: 350
Agent 7 sold a ticket. Remaining: 349
Agent 4 sold a ticket. Remaining: 348
Agent 7 sold a ticket. Remaining: 347
Agent 4 sold a ticket. Remaining: 346
Agent 7 sold a ticket. Remaining: 345
Agent 4 sold a ticket. Remaining: 344
Agent 7 sold a ticket. Remaining: 343
Agent 7 sold a ticket. Remaining: 341
Agent 3 sold a ticket. Remaining: 359
Agent 3 sold a ticket. Remaining: 340
Agent 6 sold a ticket. Remaining: 391
Agent 3 sold a ticket. Remaining: 338
Agent 3 sold a ticket. Remaining: 337
Agent 2 sold a ticket. Remaining: 354
Agent 9 sold a ticket. Remaining: 356
Agent 2 sold a ticket. Remaining: 334
Agent 8 sold a ticket. Remaining: 373
Agent 7 sold a ticket. Remaining: 339
Agent 3 sold a ticket. Remaining: 335
Agent 6 sold a ticket. Remaining: 336
Agent 6 sold a ticket. Remaining: 328
Agent 2 sold a ticket. Remaining: 332
Agent 10 sold a ticket. Remaining: 355
Agent 8 sold a ticket. Remaining: 331
Agent 10 sold a ticket. Remaining: 325
Agent 4 sold a ticket. Remaining: 342
Agent 10 sold a ticket. Remaining: 323
Agent 9 sold a ticket. Remaining: 333
Agent 10 sold a ticket. Remaining: 321
Agent 9 sold a ticket. Remaining: 320
Agent 10 sold a ticket. Remaining: 319
Agent 5 sold a ticket. Remaining: 396
Agent 10 sold a ticket. Remaining: 317
Agent 5 sold a ticket. Remaining: 316
Agent 10 sold a ticket. Remaining: 315
Agent 5 sold a ticket. Remaining: 314
Agent 3 sold a ticket. Remaining: 329
```

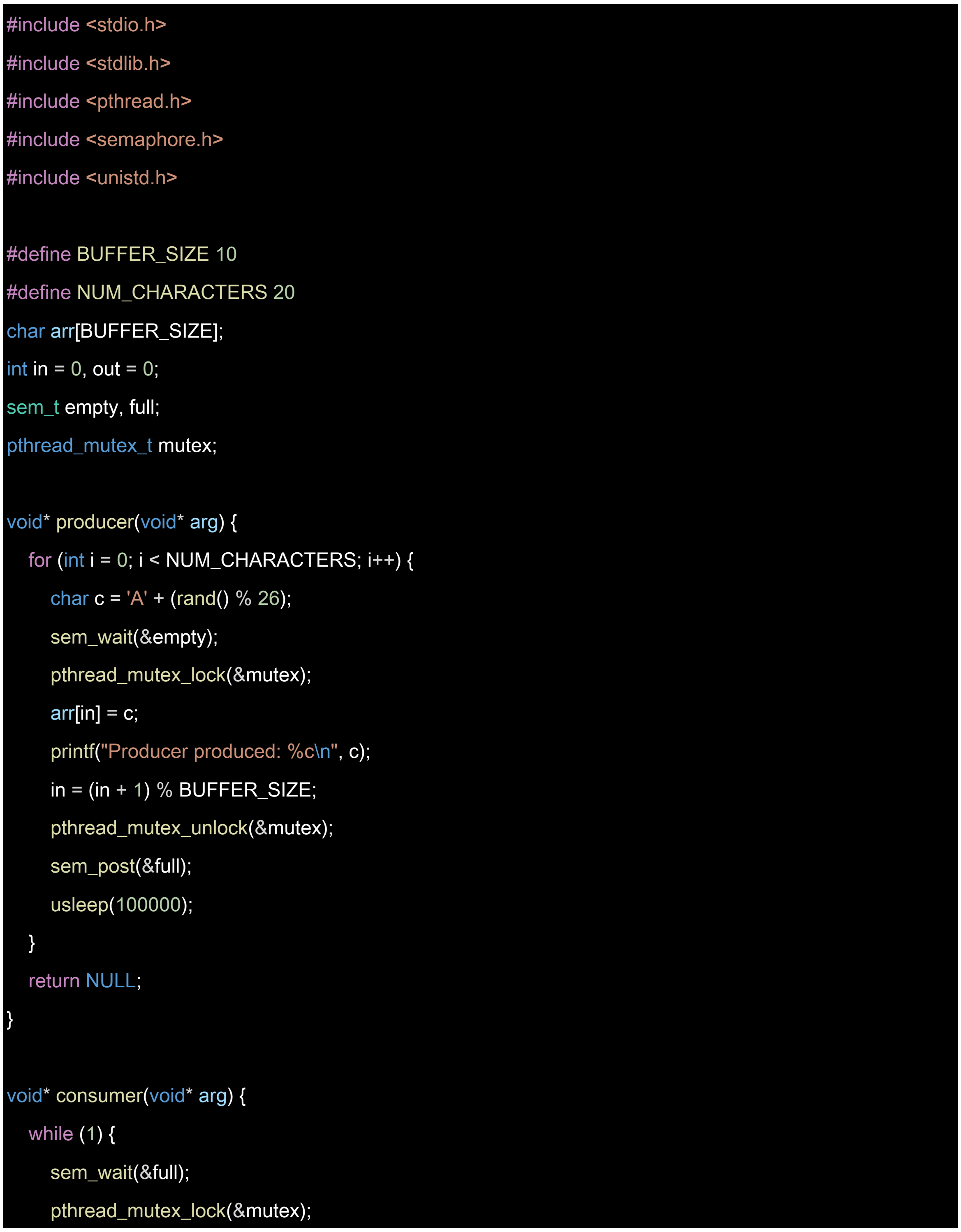
… …



PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL

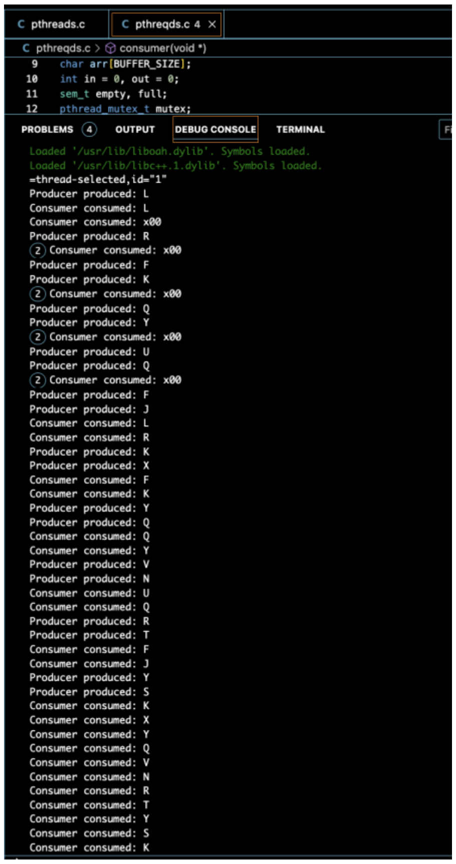
```
Agent 10 sold a ticket. Remaining: 68
Agent 10 sold a ticket. Remaining: 67
Agent 10 sold a ticket. Remaining: 66
Agent 1 sold a ticket. Remaining: 100
Agent 1 sold a ticket. Remaining: 64
Agent 4 sold a ticket. Remaining: 82
Agent 5 sold a ticket. Remaining: 97
Agent 8 sold a ticket. Remaining: 80
Agent 9 sold a ticket. Remaining: 79
Agent 9 sold a ticket. Remaining: 59
Agent 9 sold a ticket. Remaining: 58
Agent 6 sold a ticket. Remaining: 75
Agent 2 sold a ticket. Remaining: 74
Agent 2 sold a ticket. Remaining: 55
Agent 2 sold a ticket. Remaining: 54
Agent 2 sold a ticket. Remaining: 53
Agent 2 sold a ticket. Remaining: 52
Agent 2 sold a ticket. Remaining: 51
Agent 2 sold a ticket. Remaining: 50
Agent 2 sold a ticket. Remaining: 49
Agent 2 sold a ticket. Remaining: 48
Agent 2 sold a ticket. Remaining: 47
Agent 2 sold a ticket. Remaining: 46
Agent 2 sold a ticket. Remaining: 45
Agent 2 sold a ticket. Remaining: 44
Agent 2 sold a ticket. Remaining: 43
Agent 2 sold a ticket. Remaining: 42
Agent 2 sold a ticket. Remaining: 41
Agent 2 sold a ticket. Remaining: 40
Agent 2 sold a ticket. Remaining: 39
Agent 7 sold a ticket. Remaining: 83
Agent 1 sold a ticket. Remaining: 63
Agent 4 sold a ticket. Remaining: 62
Agent 5 sold a ticket. Remaining: 61
Agent 5 sold a ticket. Remaining: 34
Agent 3 sold a ticket. Remaining: 76
Agent 3 sold a ticket. Remaining: 32
Agent 3 sold a ticket. Remaining: 31
Agent 3 sold a ticket. Remaining: 30
Agent 3 sold a ticket. Remaining: 29
Agent 3 sold a ticket. Remaining: 28
Agent 3 sold a ticket. Remaining: 27
Agent 3 sold a ticket. Remaining: 26
Agent 3 sold a ticket. Remaining: 25
Agent 3 sold a ticket. Remaining: 24
Agent 3 sold a ticket. Remaining: 23
Agent 3 sold a ticket. Remaining: 22
Agent 2 sold a ticket. Remaining: 38
Agent 7 sold a ticket. Remaining: 37
Agent 1 sold a ticket. Remaining: 36
Agent 7 sold a ticket. Remaining: 19
Agent 4 sold a ticket. Remaining: 35
Agent 5 sold a ticket. Remaining: 33
Agent 4 sold a ticket. Remaining: 16
Agent 5 sold a ticket. Remaining: 15
Agent 5 sold a ticket. Remaining: 13
Agent 9 sold a ticket. Remaining: 57
Agent 6 sold a ticket. Remaining: 56
Agent 6 sold a ticket. Remaining: 10
Agent 6 sold a ticket. Remaining: 9
Agent 6 sold a ticket. Remaining: 8
Agent 6 sold a ticket. Remaining: 7
Agent 6 sold a ticket. Remaining: 6
Agent 6 sold a ticket. Remaining: 5
Agent 6 sold a ticket. Remaining: 4
Agent 6 sold a ticket. Remaining: 3
Agent 6 sold a ticket. Remaining: 2
Agent 6 sold a ticket. Remaining: 1
Agent 3 sold a ticket. Remaining: 21
Agent 2 sold a ticket. Remaining: 20
Agent 7 sold a ticket. Remaining: 17
Agent 8 sold a ticket. Remaining: 60
Agent 4 sold a ticket. Remaining: 14
Agent 5 sold a ticket. Remaining: 12
Agent 9 sold a ticket. Remaining: 11
Agent 10 sold a ticket. Remaining: 65
Agent 6 sold a ticket. Remaining: 0
Agent 1 sold a ticket. Remaining: 18
All tickets have been sold!
The program '/Users/khadichabonuvalieva/Desktop/csc408_final/pthreads' has exited with code 0 (0x00000000).
```

13. In a character array arr[10], one producer inputs random characters and two consumers consumes the characters. Write a code such that there is no race condition of two consumers and either consumer consumes characters only when they are produced by the producer. (Marks 10)



```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>

#define BUFFER_SIZE 10
#define NUM_CHARACTERS 20
char arr[BUFFER_SIZE];
int in = 0, out = 0;
sem_t empty, full;
pthread_mutex_t mutex;

void* producer(void* arg) {
    for (int i = 0; i < NUM_CHARACTERS; i++) {
        char c = 'A' + (rand() % 26);
        sem_wait(&empty);
        pthread_mutex_lock(&mutex);
        arr[in] = c;
        printf("Producer produced: %c\n", c);
        in = (in + 1) % BUFFER_SIZE;
        pthread_mutex_unlock(&mutex);
        sem_post(&full);
        usleep(100000);
    }
    return NULL;
}

void* consumer(void* arg) {
    while (1) {
        sem_wait(&full);
        pthread_mutex_lock(&mutex);
        char c = arr[out];
        printf("Consumer consumed: %c\n", c);
        out = (out + 1) % BUFFER_SIZE;
        pthread_mutex_unlock(&mutex);
        sem_post(&empty);
        usleep(200000);
    }
    return NULL;
}

int main() {
    pthread_t prod, cons1, cons2;
    sem_init(&empty, 0, BUFFER_SIZE);
    sem_init(&full, 0, 0);
    pthread_mutex_init(&mutex, NULL);
    pthread_create(&prod, NULL, producer, NULL);
    pthread_create(&cons1, NULL, consumer, NULL);
    pthread_create(&cons2, NULL, consumer, NULL);
    pthread_join(prod, NULL);
    pthread_join(cons1, NULL);
    pthread_join(cons2, NULL);
    sem_destroy(&empty);
    sem_destroy(&full);
    pthread_mutex_destroy(&mutex);
    return 0;
}
```

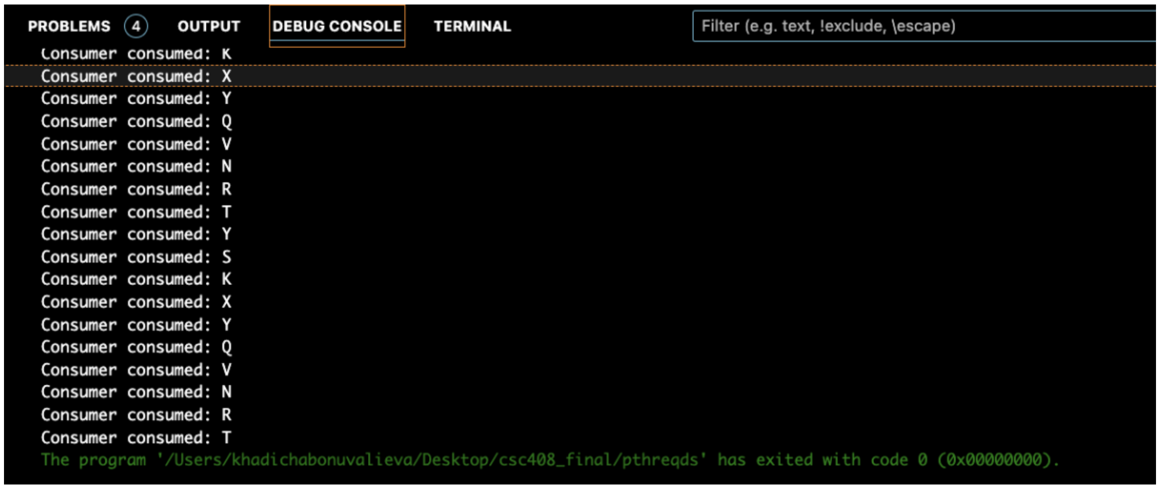
Results:



```
char arr[BUFFER_SIZE];
int in = 0, out = 0;
sem_t empty, full;
pthread_mutex_t mutex;
```

```
Loaded '/usr/lib/liboah.dylib'. Symbols loaded.
Loaded '/usr/lib/libc++.1.dylib'. Symbols loaded.
=thread-selected,id="1"
Producer produced: L
Consumer consumed: L
Consumer consumed: x00
Producer produced: R
2 Consumer consumed: x00
Producer produced: F
Producer produced: K
2 Consumer consumed: x00
Producer produced: Q
Producer produced: Y
2 Consumer consumed: x00
Producer produced: U
Producer produced: Q
2 Consumer consumed: x00
Producer produced: F
Producer produced: J
Consumer consumed: L
Consumer consumed: R
Producer produced: K
Producer produced: X
Consumer consumed: F
Consumer consumed: K
Producer produced: Y
Producer produced: Q
Consumer consumed: Q
Consumer consumed: Y
Producer produced: V
Producer produced: N
Consumer consumed: U
Consumer consumed: Q
Producer produced: R
Producer produced: T
Consumer consumed: F
Consumer consumed: J
Producer produced: Y
Producer produced: S
Consumer consumed: K
Consumer consumed: X
Consumer consumed: Y
Consumer consumed: Q
Consumer consumed: V
Consumer consumed: N
Consumer consumed: R
Consumer consumed: T
Consumer consumed: Y
Consumer consumed: S
Consumer consumed: K
```

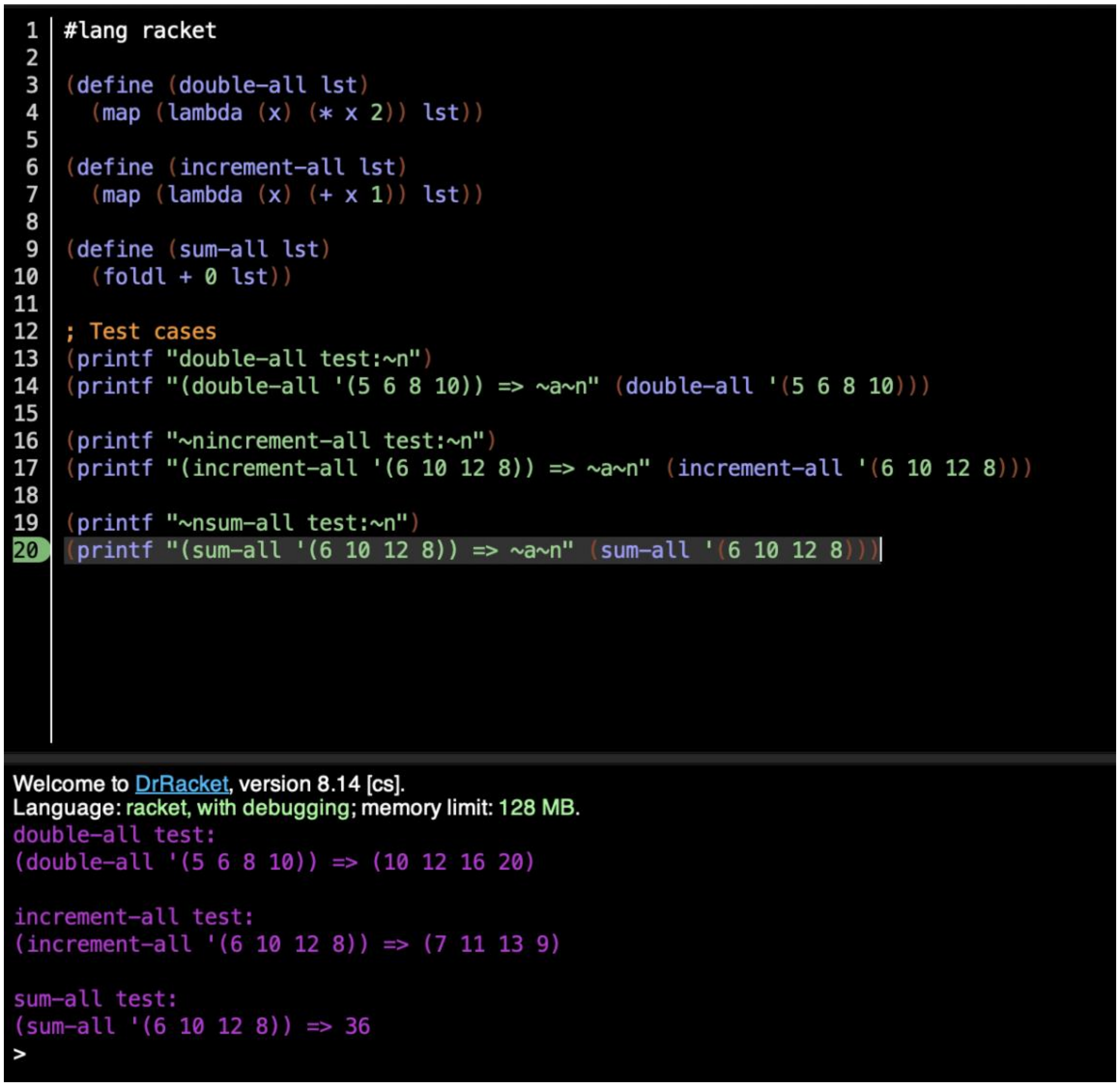
. . . .



```
Consumer consumed: K
Consumer consumed: X
Consumer consumed: Y
Consumer consumed: Q
Consumer consumed: V
Consumer consumed: N
Consumer consumed: R
Consumer consumed: T
Consumer consumed: Y
Consumer consumed: S
Consumer consumed: K
Consumer consumed: X
Consumer consumed: Y
Consumer consumed: Q
Consumer consumed: V
Consumer consumed: N
Consumer consumed: R
Consumer consumed: T
The program '/Users/khadichabonuvalieva/Desktop/csc408_final/pthreqds' has exited with code 0 (0x00000000).
```

14. Write each of the following functions (double-all, increment-all) in functional programming paradigm which when called in the prompt get the output as shown. (Marks 10)

```
> double-all '(5 6 8 10)
outputs (10 12 16 20)

>increment-all '(6 10 12 8)
outputs (7 11 13 9)
>sum-all '(6 10 12 8)
outputs 36
```



```
#lang racket

(define (double-all lst)
  (map (lambda (x) (* x 2)) lst))

(define (increment-all lst)
  (map (lambda (x) (+ x 1)) lst))

(define (sum-all lst)
  (foldl + 0 lst))

; Test cases
(printf "double-all test:~n")
(printf "(double-all '(5 6 8 10)) => ~a~n" (double-all '(5 6 8 10)))

(printf "~nincrement-all test:~n")
(printf "(increment-all '(6 10 12 8)) => ~a~n" (increment-all '(6 10 12 8)))

(printf "~nsum-all test:~n")
(printf "(sum-all '(6 10 12 8)) => ~a~n" (sum-all '(6 10 12 8)))
```

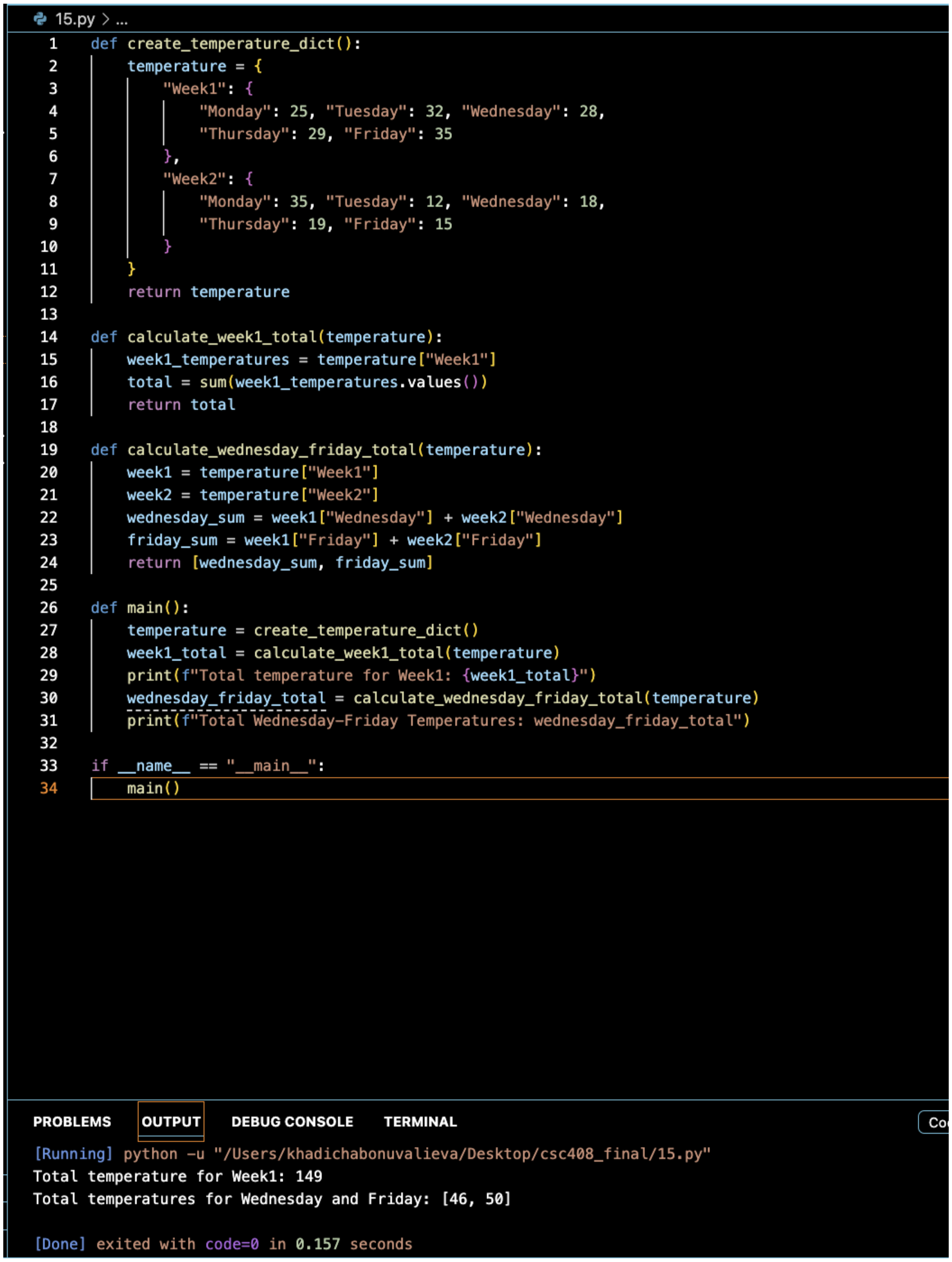
```
Welcome to DrRacket, version 8.14 [cs].
Language: racket, with debugging; memory limit: 128 MB.
double-all test:
(double-all '(5 6 8 10)) => (10 12 16 20)

increment-all test:
(increment-all '(6 10 12 8)) => (7 11 13 9)

sum-all test:
(sum-all '(6 10 12 8)) => 36
>
```

15. Create a python function to add values to an empty dictionary called temperature such that temperature ={“Week1”: {“Monday”: 25, “Tuesday”: 32, “Wednesday”: 28, “Thursday” : 29, “Friday”: 35}, Week2”: {“Monday”: 35, “Tuesday”: 12, “Wednesday”: 18, “Thursday” : 19, “Friday”: 15}}
Create another two functions, the first one adds all the temperatures of Week1 and outputs the added value, and the second one adds the temperatures of Wednesday of Week1 and Week2, and add the temperatures of Friday of Week1 and Week2 and output it as a list (Marks 10)

```
def create_temperature_dict():
```

```python
    temperature = {
        "Week1": {
            "Monday": 25, "Tuesday": 32, "Wednesday": 28,
            "Thursday": 29, "Friday": 35
        },
        "Week2": {
            "Monday": 35, "Tuesday": 12, "Wednesday": 18,
            "Thursday": 19, "Friday": 15
        }
    }
    return temperature

def calculate_week1_total(temperature):
    week1_temperatures = temperature["Week1"]
    total = sum(week1_temperatures.values())
    return total

def calculate_wednesday_friday_total(temperature):
    week1 = temperature["Week1"]
    week2 = temperature["Week2"]
    wednesday_sum = week1["Wednesday"] + week2["Wednesday"]
    friday_sum = week1["Friday"] + week2["Friday"]
    return [wednesday_sum, friday_sum]

def main():
    temperature = create_temperature_dict()
    week1_total = calculate_week1_total(temperature)
    print(f"Total temperature for Week1: {week1_total}")
    wednesday_friday_total = calculate_wednesday_friday_total(temperature)
    print(f"Total Wednesday-Friday Temperatures: wednesday_friday_total")

if __name__ == "__main__":
    main()
```

Result:



15.py > ...

```python
def create_temperature_dict():
    temperature = {
        "Week1": {
            "Monday": 25, "Tuesday": 32, "Wednesday": 28,
            "Thursday": 29, "Friday": 35
        },
        "Week2": {
            "Monday": 35, "Tuesday": 12, "Wednesday": 18,
            "Thursday": 19, "Friday": 15
        }
    }
    return temperature

def calculate_week1_total(temperature):
    week1_temperatures = temperature["Week1"]
    total = sum(week1_temperatures.values())
    return total

def calculate_wednesday_friday_total(temperature):
    week1 = temperature["Week1"]
    week2 = temperature["Week2"]
    wednesday_sum = week1["Wednesday"] + week2["Wednesday"]
    friday_sum = week1["Friday"] + week2["Friday"]
    return [wednesday_sum, friday_sum]

def main():
    temperature = create_temperature_dict()
    week1_total = calculate_week1_total(temperature)
    print(f"Total temperature for Week1: {week1_total}")
    wednesday_friday_total = calculate_wednesday_friday_total(temperature)
    print(f"Total Wednesday-Friday Temperatures: wednesday_friday_total")

if __name__ == "__main__":
    main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
[Running] python -u "/Users/khadichabonuvalieva/Desktop/csc408_final/15.py"
Total temperature for Week1: 149
Total temperatures for Wednesday and Friday: [46, 50]

[Done] exited with code=0 in 0.157 seconds
```