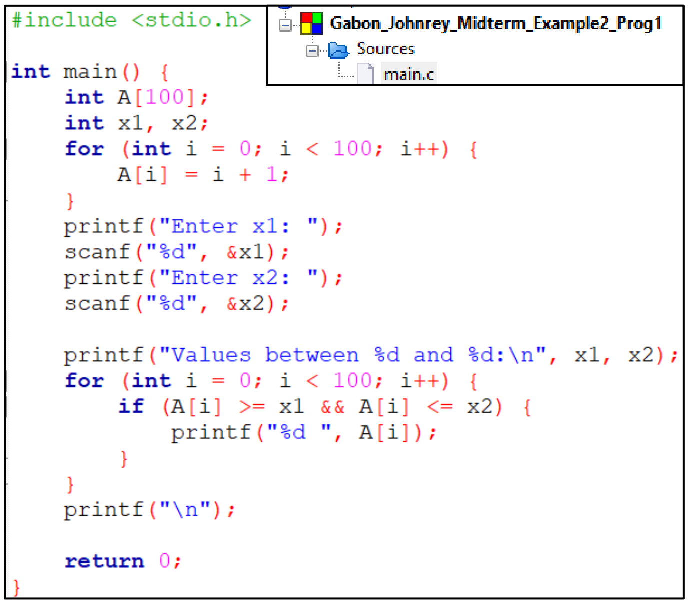
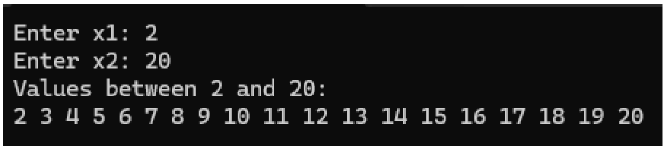
MIDTERM EXAM EXAMPLE 2

PROBLEM 1:

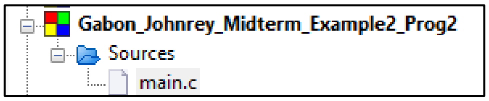


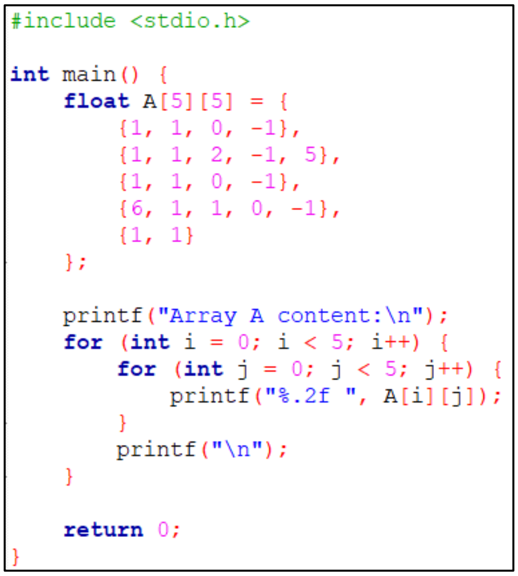
```c
#include <stdio.h>

int main() {
    int A[100];
    int x1, x2;
    for (int i = 0; i < 100; i++) {
        A[i] = i + 1;
    }
    printf("Enter x1: ");
    scanf("%d", &x1);
    printf("Enter x2: ");
    scanf("%d", &x2);

    printf("Values between %d and %d:\n", x1, x2);
    for (int i = 0; i < 100; i++) {
        if (A[i] >= x1 && A[i] <= x2) {
            printf("%d ", A[i]);
        }
    }
    printf("\n");

    return 0;
}
```



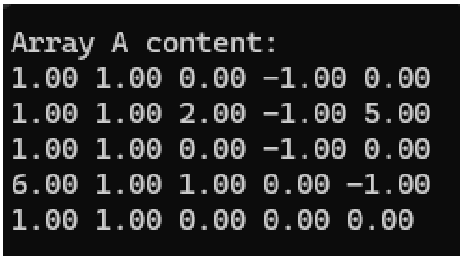
```
Enter x1: 2
Enter x2: 20
Values between 2 and 20:
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

PROBLEM 2:



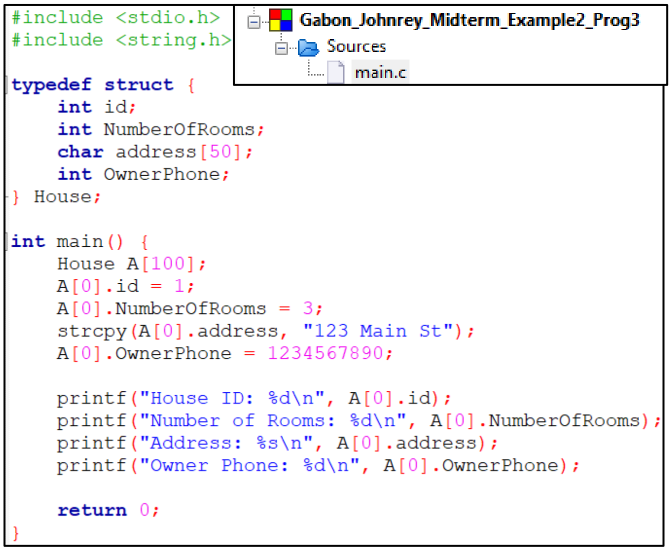


```c
#include <stdio.h>

int main() {
    float A[5][5] = {
        {1, 1, 0, -1},
        {1, 1, 2, -1, 5},
        {1, 1, 0, -1},
        {6, 1, 1, 0, -1},
        {1, 1}
    };

    printf("Array A content:\n");
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 5; j++) {
            printf("%.2f ", A[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

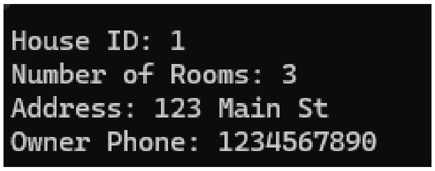


```
Array A content:
1.00 1.00 0.00 -1.00 0.00
1.00 1.00 2.00 -1.00 5.00
1.00 1.00 0.00 -1.00 0.00
6.00 1.00 1.00 0.00 -1.00
1.00 1.00 0.00 0.00 0.00
```

PROBLEM 3:

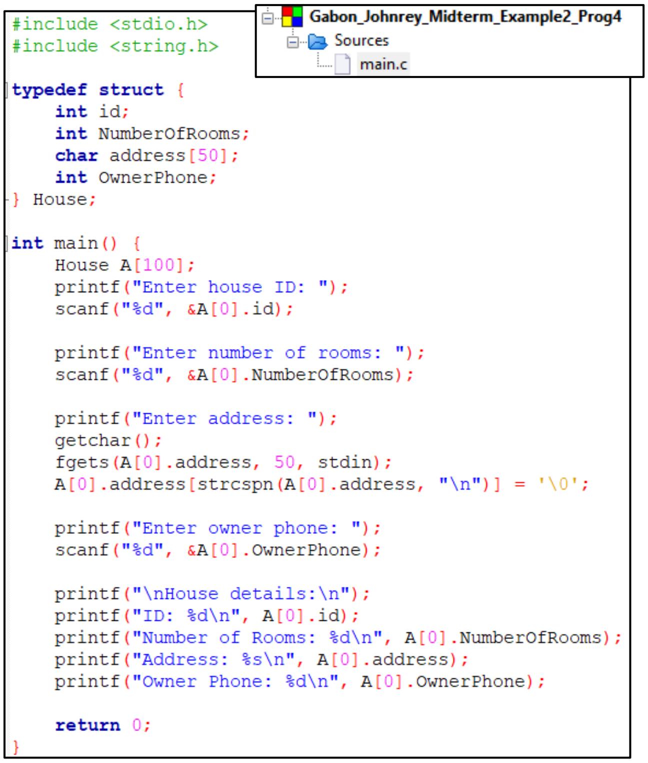


```c
#include <stdio.h>
#include <string.h>

typedef struct {
    int id;
    int NumberOfRooms;
    char address[50];
    int OwnerPhone;
} House;

int main() {
    House A[100];
    A[0].id = 1;
    A[0].NumberOfRooms = 3;
    strcpy(A[0].address, "123 Main St");
    A[0].OwnerPhone = 1234567890;

    printf("House ID: %d\n", A[0].id);
    printf("Number of Rooms: %d\n", A[0].NumberOfRooms);
    printf("Address: %s\n", A[0].address);
    printf("Owner Phone: %d\n", A[0].OwnerPhone);

    return 0;
}
```

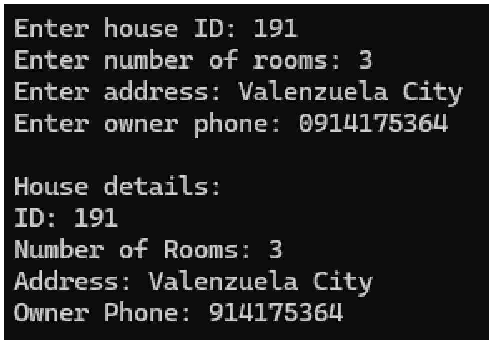


```
House ID: 1
Number of Rooms: 3
Address: 123 Main St
Owner Phone: 1234567890
```

PROBLEM 4:

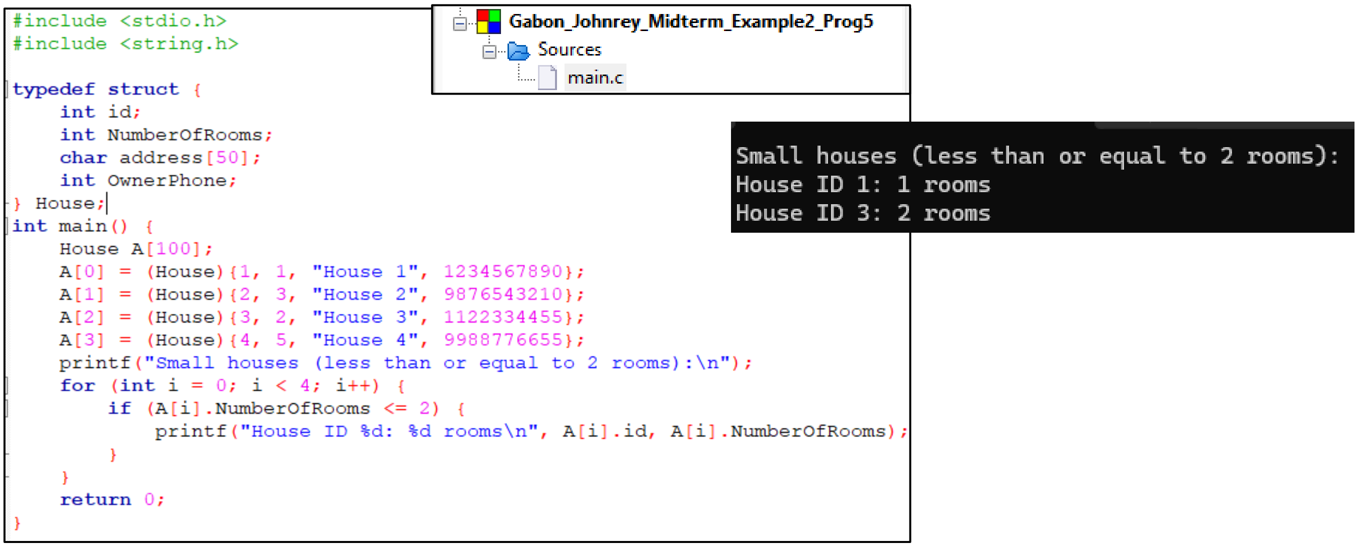


```c
#include <stdio.h>
#include <string.h>

typedef struct {
    int id;
    int NumberOfRooms;
    char address[50];
    int OwnerPhone;
} House;

int main() {
    House A[100];
    printf("Enter house ID: ");
    scanf("%d", &A[0].id);

    printf("Enter number of rooms: ");
    scanf("%d", &A[0].NumberOfRooms);

    printf("Enter address: ");
    getchar();
    fgets(A[0].address, 50, stdin);
    A[0].address[strcspn(A[0].address, "\n")] = '\0';

    printf("Enter owner phone: ");
    scanf("%d", &A[0].OwnerPhone);

    printf("\nHouse details:\n");
    printf("ID: %d\n", A[0].id);
    printf("Number of Rooms: %d\n", A[0].NumberOfRooms);
    printf("Address: %s\n", A[0].address);
    printf("Owner Phone: %d\n", A[0].OwnerPhone);

    return 0;
}
```



```
Enter house ID: 191
Enter number of rooms: 3
Enter address: Valenzuela City
Enter owner phone: 0914175364

House details:
ID: 191
Number of Rooms: 3
Address: Valenzuela City
Owner Phone: 914175364
```

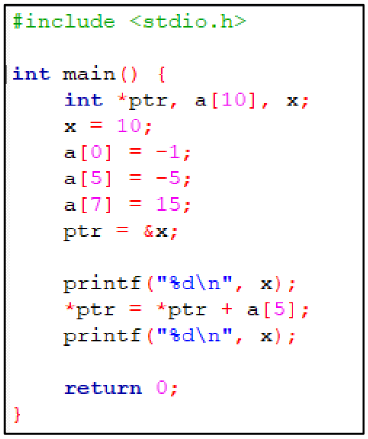
PROBLEM 5:



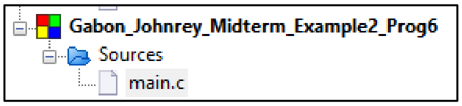
```c
#include <stdio.h>
#include <string.h>

typedef struct {
    int id;
    int NumberOfRooms;
    char address[50];
    int OwnerPhone;
} House;
int main() {
    House A[100];
    A[0] = (House){1, 1, "House 1", 1234567890};
    A[1] = (House){2, 3, "House 2", 9876543210};
    A[2] = (House){3, 2, "House 3", 1122334455};
    A[3] = (House){4, 5, "House 4", 9988776655};
    printf("Small houses (less than or equal to 2 rooms):\n");
    for (int i = 0; i < 4; i++) {
        if (A[i].NumberOfRooms <= 2) {
            printf("House ID %d: %d rooms\n", A[i].id, A[i].NumberOfRooms);
        }
    }
    return 0;
}
```

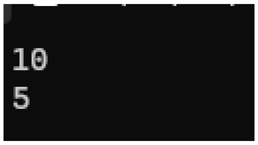
```
Small houses (less than or equal to 2 rooms):
House ID 1: 1 rooms
House ID 3: 2 rooms
```

PROBLEM 6:



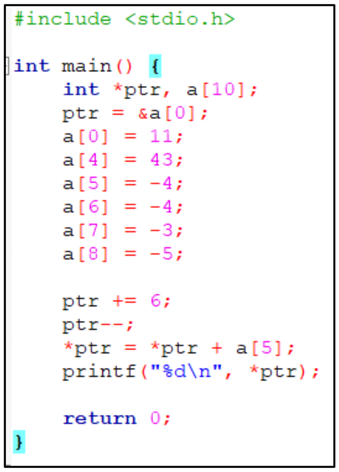
```c
#include <stdio.h>

int main() {
    int *ptr, a[10], x;
    x = 10;
    a[0] = -1;
    a[5] = -5;
    a[7] = 15;
    ptr = &x;

    printf("%d\n", x);
    *ptr = *ptr + a[5];
    printf("%d\n", x);

    return 0;
}
```



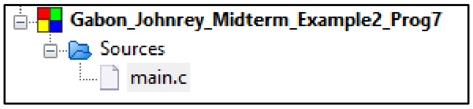


```
10
5
```

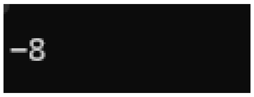
PROBLEM 7:



```c
#include <stdio.h>

int main() {
    int *ptr, a[10];
    ptr = &a[0];
    a[0] = 11;
    a[4] = 43;
    a[5] = -4;
    a[6] = -4;
    a[7] = -3;
    a[8] = -5;

    ptr += 6;
    ptr--;
    *ptr = *ptr + a[5];
    printf("%d\n", *ptr);

    return 0;
}
```

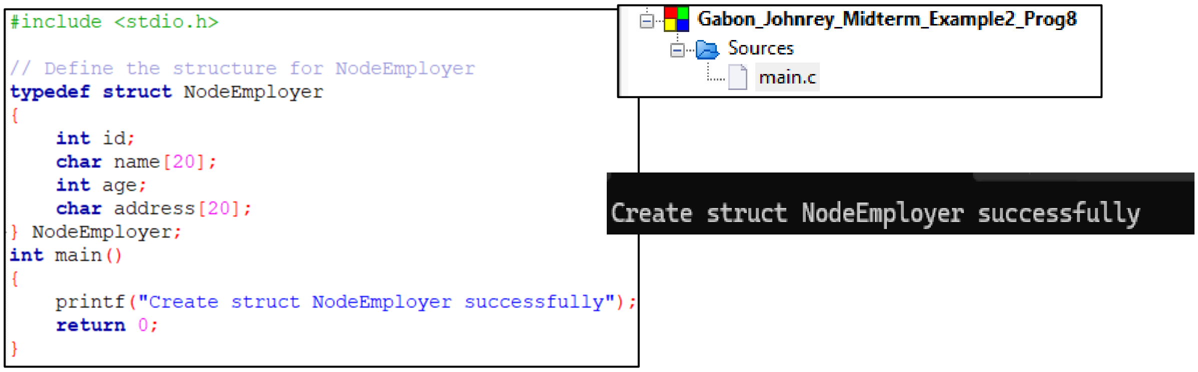


```
Gabon_Johnrey_Midterm_Example2_Prog7
    Sources
        main.c
```



```
-8
```

PROBLEM 8:

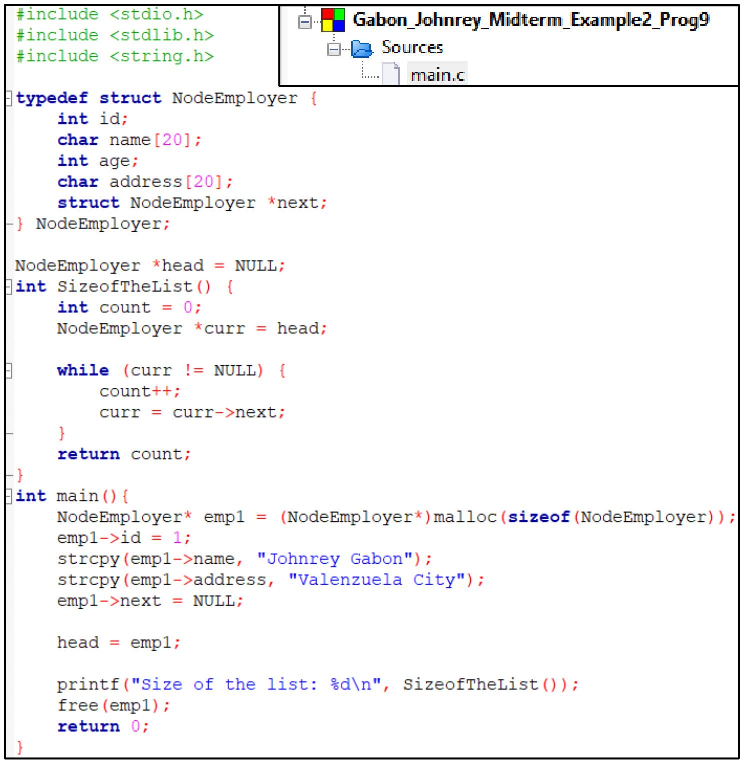


```c
#include <stdio.h>

// Define the structure for NodeEmployer
typedef struct NodeEmployer
{
    int id;
    char name[20];
    int age;
    char address[20];
} NodeEmployer;
int main()
{
    printf("Create struct NodeEmployer successfully");
    return 0;
}
```

```
Gabon_Johnrey_Midterm_Example2_Prog8
    Sources
        main.c
```

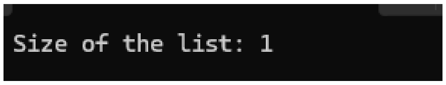
```
Create struct NodeEmployer successfully
```

PROBLEM 9:



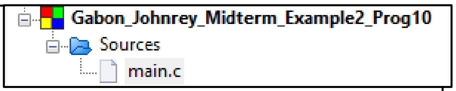
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct NodeEmployer {
    int id;
    char name[20];
    int age;
    char address[20];
    struct NodeEmployer *next;
} NodeEmployer;

NodeEmployer *head = NULL;
int SizeofTheList() {
    int count = 0;
    NodeEmployer *curr = head;

    while (curr != NULL) {
        count++;
        curr = curr->next;
    }
    return count;
}
int main(){
    NodeEmployer* emp1 = (NodeEmployer*)malloc(sizeof(NodeEmployer));
    emp1->id = 1;
    strcpy(emp1->name, "Johnrey Gabon");
    strcpy(emp1->address, "Valenzuela City");
    emp1->next = NULL;

    head = emp1;

    printf("Size of the list: %d\n", SizeofTheList());
    free(emp1);
    return 0;
}
```

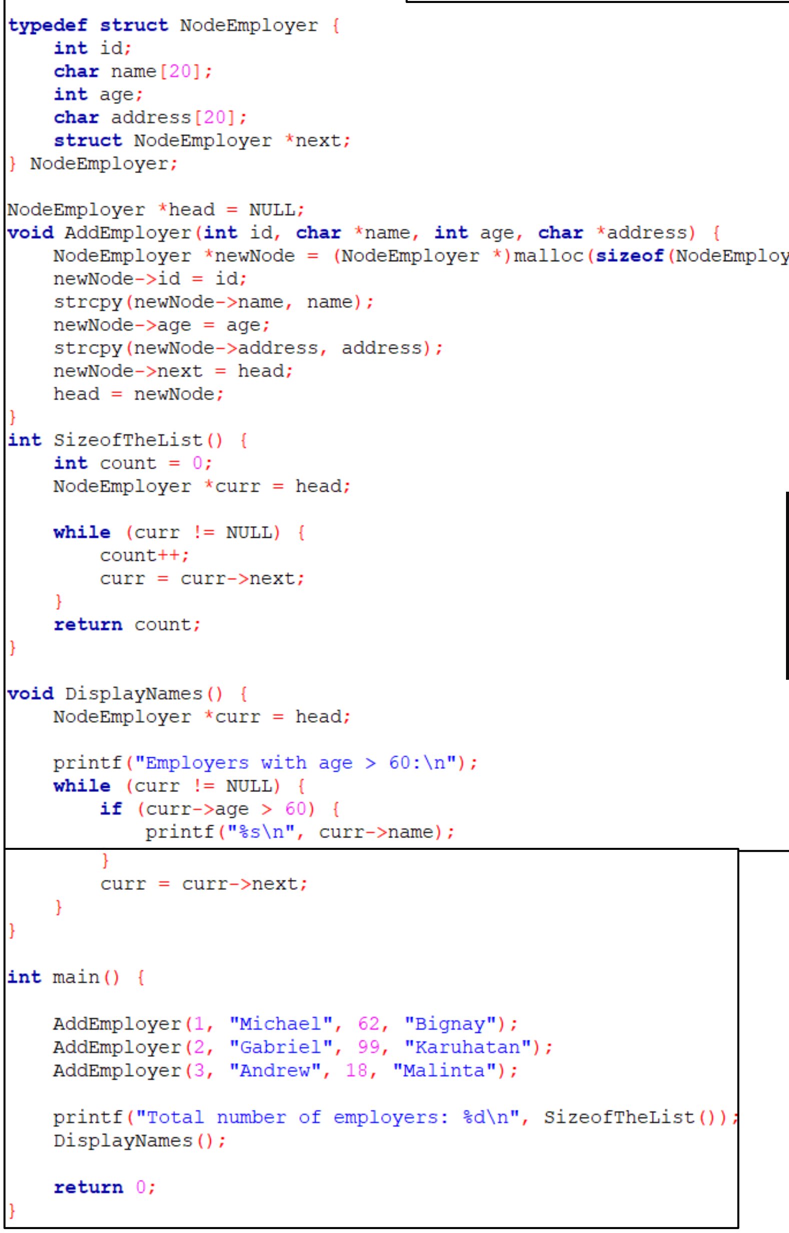
```
Gabon_Johnrey_Midterm_Example2_Prog9
    Sources
        main.c
```



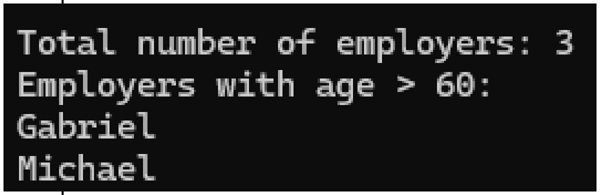
```
Size of the list: 1
```

PROBLEM 10:





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

typedef struct NodeEmployer {
    int id;
    char name[20];
    int age;
    char address[20];
    struct NodeEmployer *next;
} NodeEmployer;

NodeEmployer *head = NULL;
void AddEmployer(int id, char *name, int age, char *address) {
    NodeEmployer *newNode = (NodeEmployer *)malloc(sizeof(NodeEmployer));
    newNode->id = id;
    strcpy(newNode->name, name);
    newNode->age = age;
    strcpy(newNode->address, address);
    newNode->next = head;
    head = newNode;
}
int SizeofTheList() {
    int count = 0;
    NodeEmployer *curr = head;

    while (curr != NULL) {
        count++;
        curr = curr->next;
    }
    return count;
}

void DisplayNames() {
    NodeEmployer *curr = head;

    printf("Employers with age > 60:\n");
    while (curr != NULL) {
        if (curr->age > 60) {
            printf("%s\n", curr->name);
        }
        curr = curr->next;
    }
}

int main() {

    AddEmployer(1, "Michael", 62, "Bignay");
    AddEmployer(2, "Gabriel", 99, "Karuhatan");
    AddEmployer(3, "Andrew", 18, "Malinta");

    printf("Total number of employers: %d\n", SizeofTheList());
    DisplayNames();

    return 0;
}
```



```
Total number of employers: 3
Employers with age > 60:
Gabriel
Michael
```