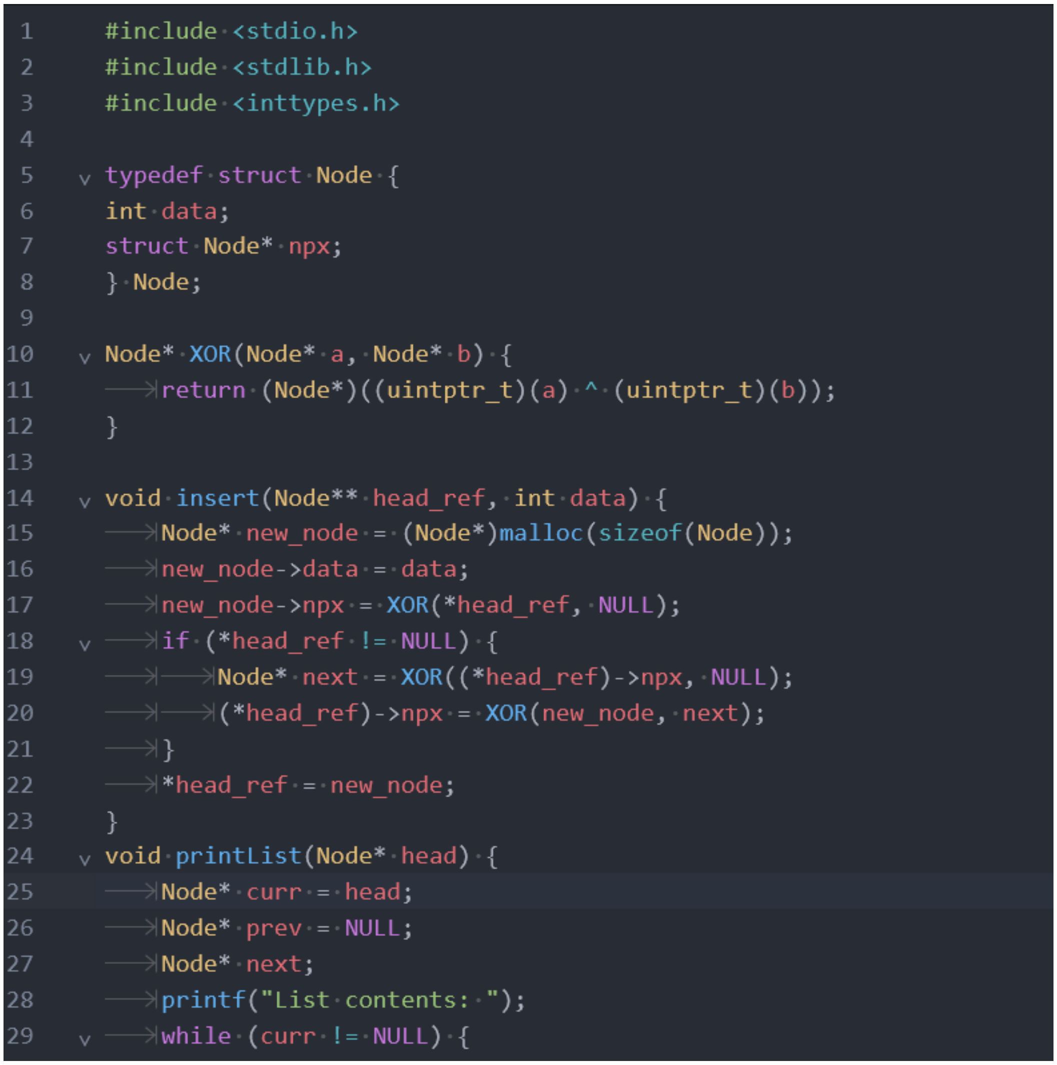
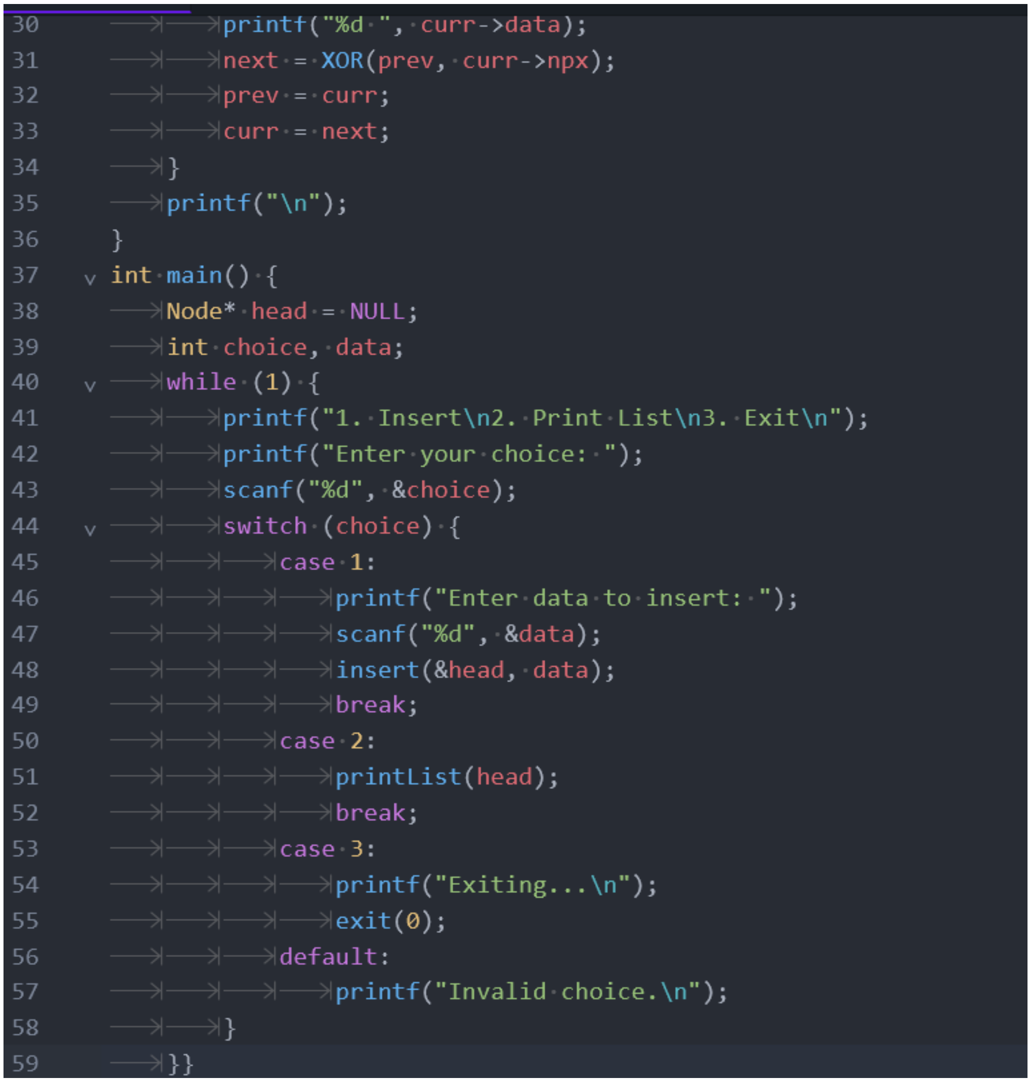
DS Lab 10 Solutions

10.1.1

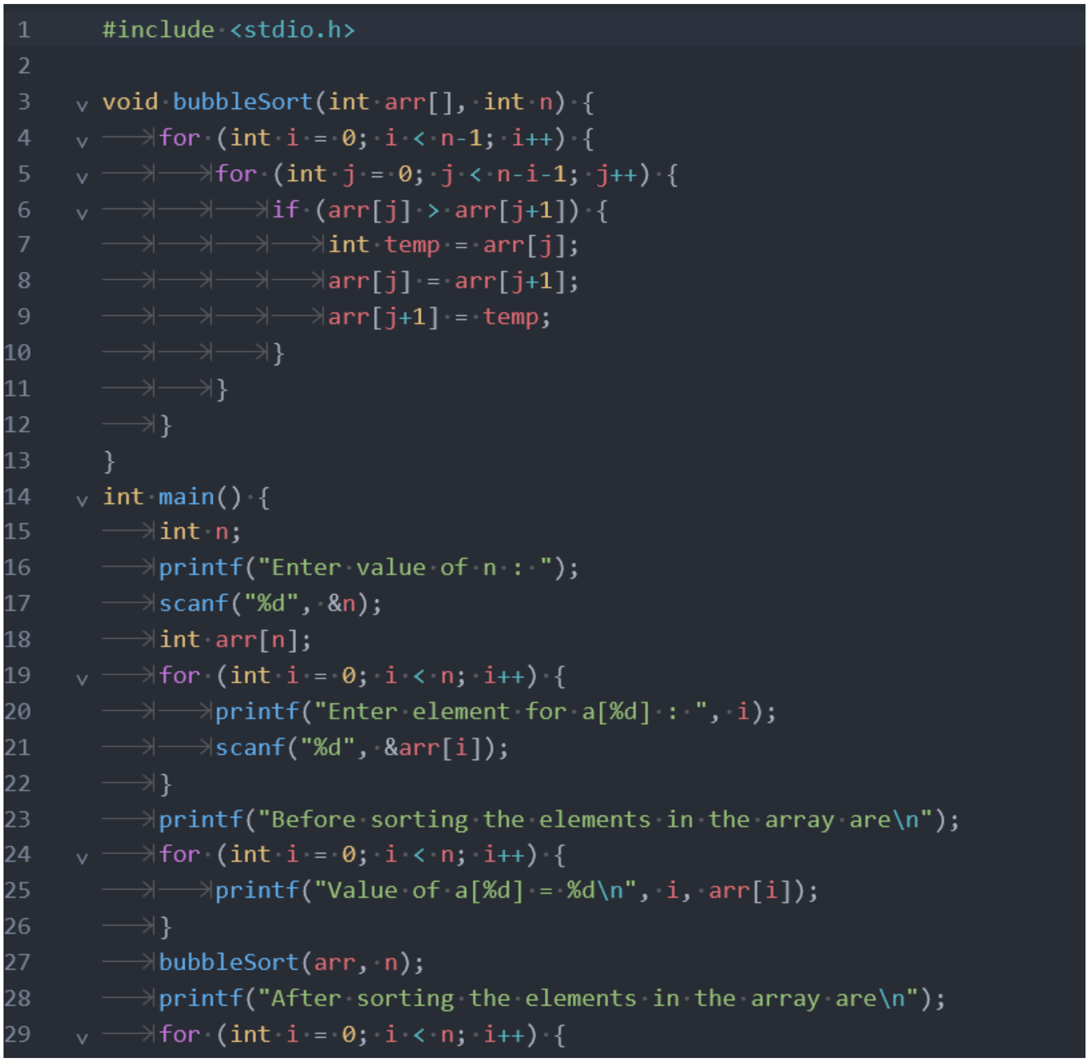


```
#include <stdio.h>
#include <stdlib.h>
#include <inttypes.h>

typedef struct Node {
int data;
struct Node* npx;
} Node;

Node* XOR(Node* a, Node* b) {
	return (Node*)((uintptr_t)(a) ^ (uintptr_t)(b));
}

void insert(Node** head_ref, int data) {
	Node* new_node = (Node*)malloc(sizeof(Node));
	new_node->data = data;
	new_node->npx = XOR(*head_ref, NULL);
	if (*head_ref != NULL) {
		Node* next = XOR((*head_ref)->npx, NULL);
		(*head_ref)->npx = XOR(new_node, next);
	}
	*head_ref = new_node;
}
void printList(Node* head) {
	Node* curr = head;
	Node* prev = NULL;
	Node* next;
	printf("List contents: ");
	while (curr != NULL) {
```

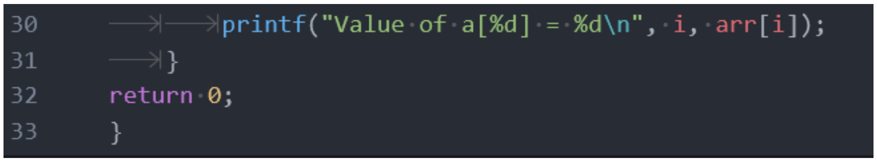


```c
        printf("%d ", curr->data);
        next = XOR(prev, curr->npx);
        prev = curr;
        curr = next;
    }
    printf("\n");
}
int main() {
    Node* head = NULL;
    int choice, data;
    while (1) {
        printf("1. Insert\n2. Print List\n3. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                printf("Enter data to insert: ");
                scanf("%d", &data);
                insert(&head, data);
                break;
            case 2:
                printList(head);
                break;
            case 3:
                printf("Exiting...\n");
                exit(0);
            default:
                printf("Invalid choice.\n");
        }
    }}
```

10.1.2

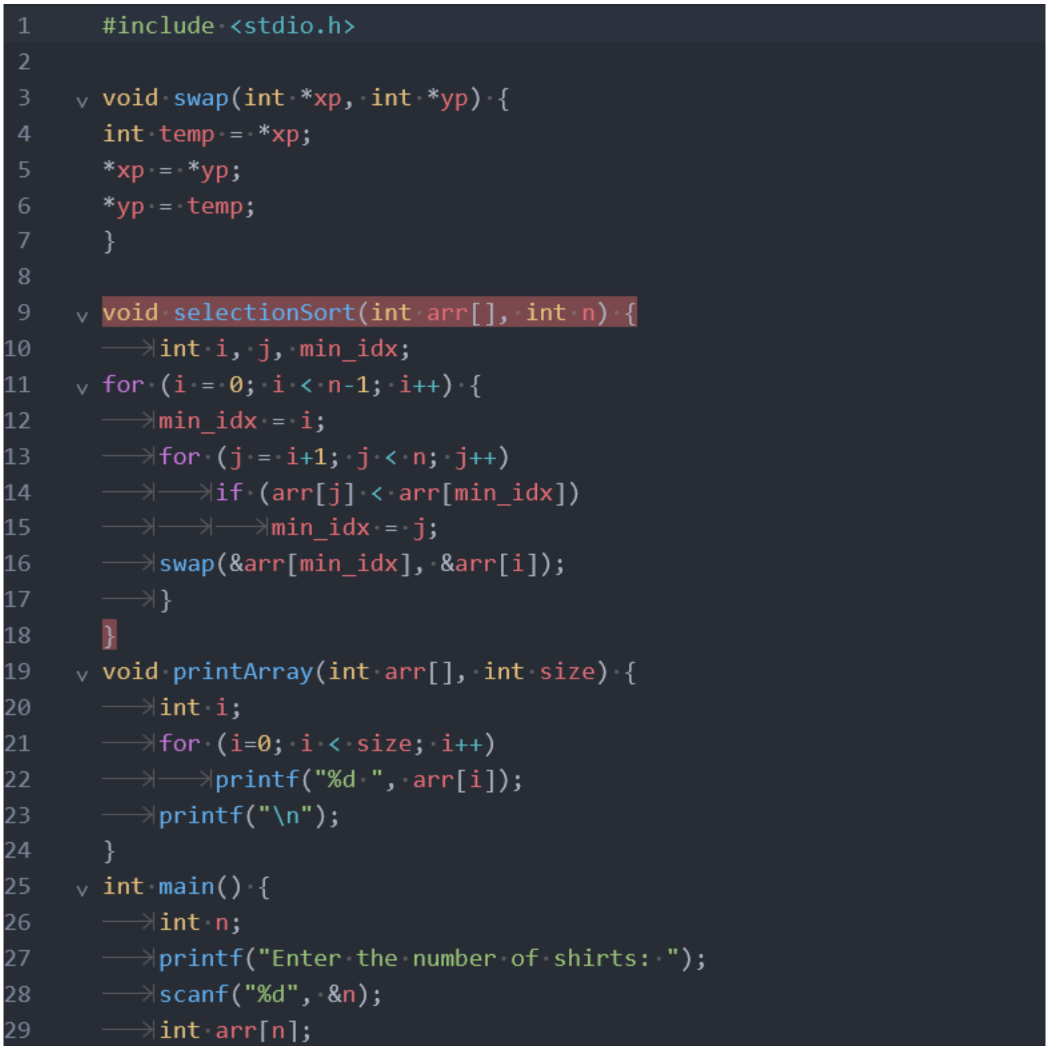


```c
#include <stdio.h>

void bubbleSort(int arr[], int n) {
    for (int i = 0; i < n-1; i++) {
        for (int j = 0; j < n-i-1; j++) {
            if (arr[j] > arr[j+1]) {
                int temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}
int main() {
    int n;
    printf("Enter value of n : ");
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++) {
        printf("Enter element for a[%d] : ", i);
        scanf("%d", &arr[i]);
    }
    printf("Before sorting the elements in the array are\n");
    for (int i = 0; i < n; i++) {
        printf("Value of a[%d] = %d\n", i, arr[i]);
    }
    bubbleSort(arr, n);
    printf("After sorting the elements in the array are\n");
    for (int i = 0; i < n; i++) {
```

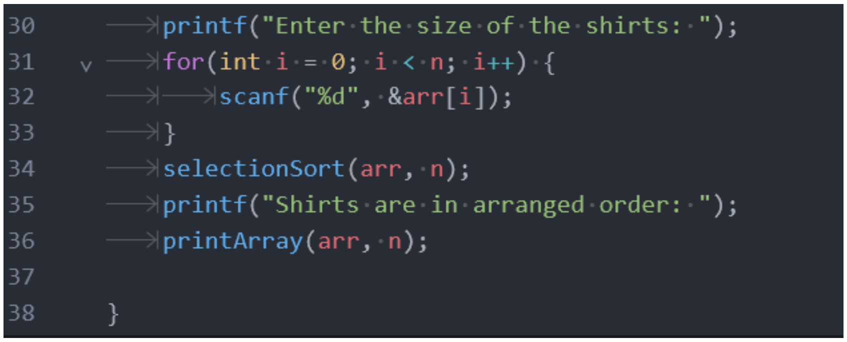


```c
        printf("Value of a[%d] = %d\n", i, arr[i]);
    }
return 0;
}
```

10.2.1

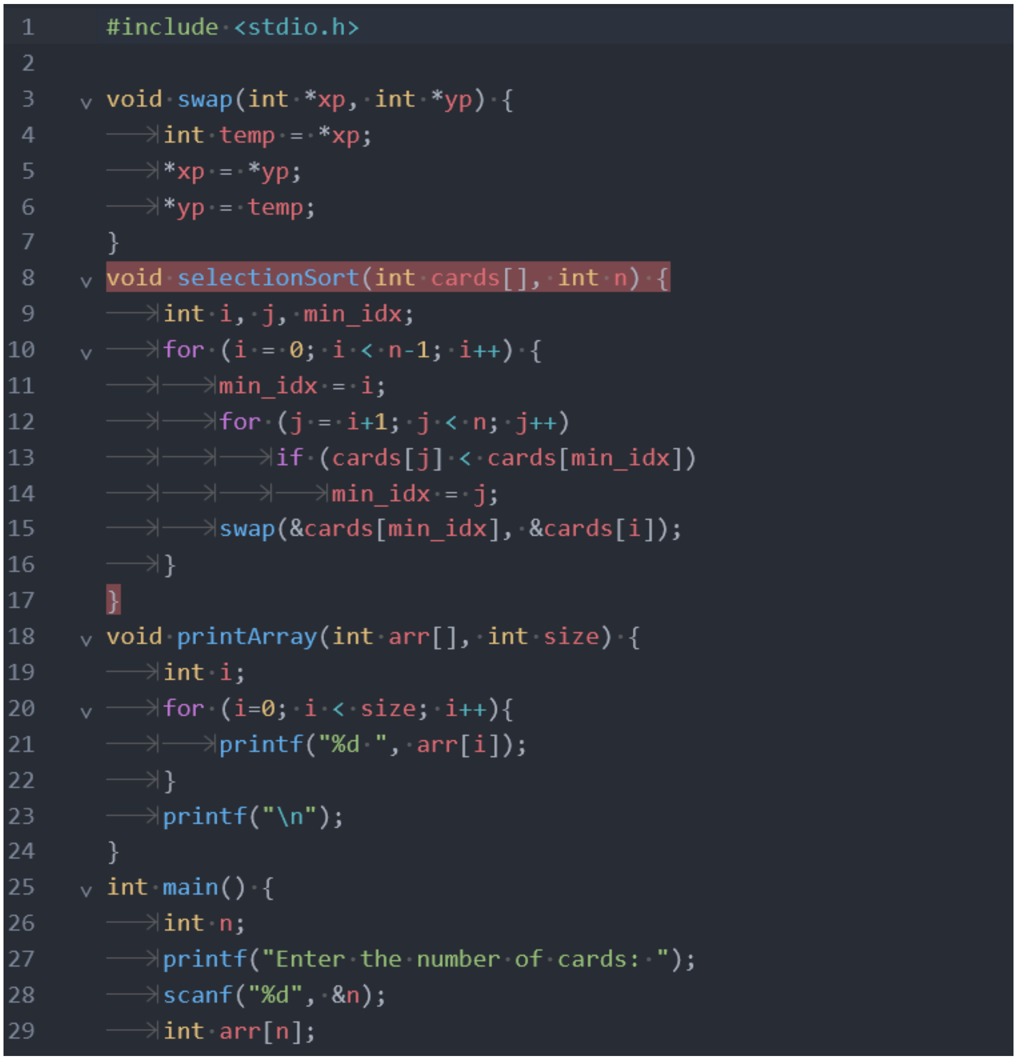


```c
#include <stdio.h>

void swap(int *xp, int *yp) {
int temp = *xp;
*xp = *yp;
*yp = temp;
}

void selectionSort(int arr[], int n) {
    int i, j, min_idx;
for (i = 0; i < n-1; i++) {
    min_idx = i;
    for (j = i+1; j < n; j++)
        if (arr[j] < arr[min_idx])
            min_idx = j;
    swap(&arr[min_idx], &arr[i]);
    }
}
void printArray(int arr[], int size) {
    int i;
    for (i=0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
}
int main() {
    int n;
    printf("Enter the number of shirts: ");
    scanf("%d", &n);
    int arr[n];
```

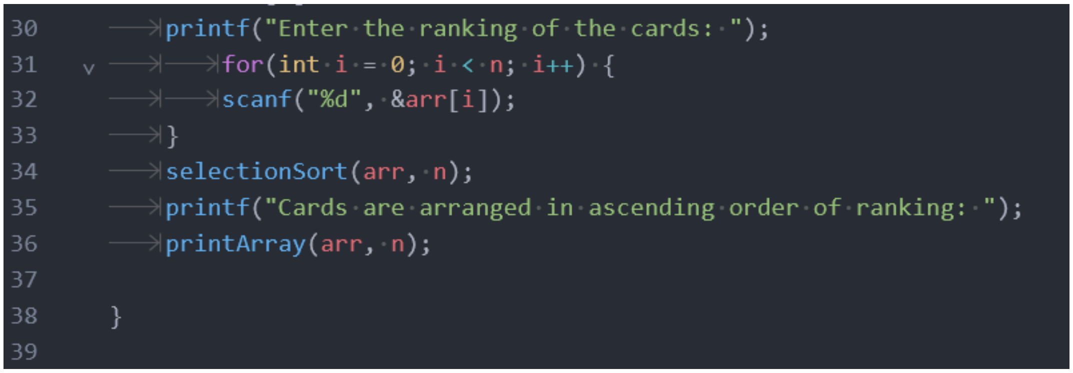


```c
    printf("Enter the size of the shirts: ");
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    selectionSort(arr, n);
    printf("Shirts are in arranged order: ");
    printArray(arr, n);

}
```

10.2.2



```c
#include <stdio.h>

void swap(int *xp, int *yp) {
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}
void selectionSort(int cards[], int n) {
    int i, j, min_idx;
    for (i = 0; i < n-1; i++) {
        min_idx = i;
        for (j = i+1; j < n; j++)
            if (cards[j] < cards[min_idx])
                min_idx = j;
        swap(&cards[min_idx], &cards[i]);
    }
}
void printArray(int arr[], int size) {
    int i;
    for (i=0; i < size; i++){
        printf("%d ", arr[i]);
    }
    printf("\n");
}
int main() {
    int n;
    printf("Enter the number of cards: ");
    scanf("%d", &n);
    int arr[n];
```



```c
    printf("Enter the ranking of the cards: ");
        for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    selectionSort(arr, n);
    printf("Cards are arranged in ascending order of ranking: ");
    printArray(arr, n);

}

```