Priority Queue Program:

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

#define MAX 5

int data[MAX];
int priority[MAX];
int size = 0;

// Insert an element into the priority queue
void insert(int value, int pr) {
    int i = size - 1;

    if (size == MAX) {
        printf("Queue is full! Cannot insert.\n");
        return;
    }

    // Shift elements to find correct position
    while (i >= 0 && priority[i] > pr) {
        data[i + 1] = data[i];
        priority[i + 1] = priority[i];
        i--;
    }

    data[i + 1] = value;
    priority[i + 1] = pr;
    size++;

    printf("Inserted value %d with priority %d.\n", value, pr);
}

// Delete the element with highest priority
void delete() {
    if (size == 0) {
        printf("Queue is empty! Nothing to delete.\n");
```

```
    printf("Inserted value %d with priority %d.\n", value, pr);
}

// Delete the element with highest priority
void delete() {
    if (size == 0) {
        printf("Queue is empty! Nothing to delete.\n");
        return;
    }

    printf("Deleted element: %d (Priority: %d)\n", data[0], priority[0]);

    // Shift remaining elements
    for (int i = 1; i < size; i++) {
        data[i - 1] = data[i];
        priority[i - 1] = priority[i];
    }

    size--;
}

// Display queue
void display() {
    if (size == 0) {
        printf("Queue is empty.\n");
        return;
    }

    printf("\nCurrent Priority Queue:\n");
    printf("Data\tPriority\n");

    for (int i = 0; i < size; i++) {
        printf("%d\t%d\n", data[i], priority[i]);
    }
}
```

```c

// Display queue
void display() {
    if (size == 0) {
        printf("Queue is empty.\n");
        return;
    }

    printf("\nCurrent Priority Queue:\n");
    printf("Data\tPriority\n");

    for (int i = 0; i < size; i++) {
        printf("%d\t%d\n", data[i], priority[i]);
    }
}

int main() {
    int choice, value, pr;

    while (1) {
        printf("\n--- PRIORITY QUEUE USING ARRAY ---\n");
        printf("1. Insert Element\n");
        printf("2. Delete Element\n");
        printf("3. Display Queue\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                if (size == MAX) {
                    printf("Queue is full! Cannot insert.\n");
                    break;
                }
                printf("Enter value: ");
                scanf("%d", &value);
```

```
Enter value: 10
Enter priority (smaller number = higher priority): 1
Inserted value 10 with priority 1.--- PRIORITY QUEUE USING ARRAY ---1. Insert Element2. Delete Element3. Display Queue4. ExitEnter your choice: 1
Enter value: 20
Enter priority (smaller number = higher priority): 2
Inserted value 20 with priority 2.--- PRIORITY QUEUE USING ARRAY ---1. Insert Element2. Delete Element3. Display Queue4. ExitEnter your choice: 1
Enter value: 30
Enter priority (smaller number = higher priority): 3
Inserted value 30 with priority 3.--- PRIORITY QUEUE USING ARRAY ---1. Insert Element2. Delete Element3. Display Queue4. ExitEnter your choice: 1
Enter value: 40
Enter priority (smaller number = higher priority): 4
Inserted value 40 with priority 4.--- PRIORITY QUEUE USING ARRAY ---1. Insert Element2. Delete Element3. Display Queue4. ExitEnter your choice: 1
Enter value: 50
Enter priority (smaller number = higher priority): 5
Inserted value 50 with priority 5.--- PRIORITY QUEUE USING ARRAY ---1. Insert Element2. Delete Element3. Display Queue4. ExitEnter your choice: 2
Deleted element: 10 (Priority: 1)--- PRIORITY QUEUE USING ARRAY ---1. Insert Element2. Delete Element3. Display Queue4. ExitEnter your choice: 2
Deleted element: 20 (Priority: 2)--- PRIORITY QUEUE USING ARRAY ---1. Insert Element2. Delete Element3. Display Queue4. ExitEnter your choice: 3
Current Priority Queue:Data     Priority30      340     450     5--- PRIORITY QUEUE USING ARRAY ---1. Insert Element2. Delete Element3. Display Queue4. Exit
Enter your choice: 4

Process returned 0 (0x0)   execution time : 28.612 s
Press any key to continue.
```