

MONEEB TOOR

221048

OS LAB 5

FINAL PAPER

Que 1

CODE

```
#include <iostream>
#include <queue>
#include <algorithm>

using namespace std;

struct Patient {
    int id;
    bool emergency;
    int age;
    int estimated_time;
    string priority_group;
};

struct EmergencyComparator {
    bool operator()(const Patient& p1, const Patient& p2) {
        return p1.estimated_time > p2.estimated_time;
    }
};

struct PriorityComparator {
    bool operator()(const Patient& p1, const Patient& p2) {
        return p1.estimated_time > p2.estimated_time;
    }
};
```

```

void processPatient(Patient patients[], int &patientCount) {
    cout << "Enter patient ID: ";
    cin >> patients[patientCount].id;

    cout << "Is this an emergency (1 for Yes, 0 for No)? ";
    cin >> patients[patientCount].emergency;

    cout << "Enter patient's age: ";
    cin >> patients[patientCount].age;

    cout << "Enter estimated consultation time (in minutes): ";
    cin >> patients[patientCount].estimated_time;

    if (patients[patientCount].emergency) {
        patients[patientCount].priority_group = "Emergency";
    } else if (patients[patientCount].age < 18) {
        patients[patientCount].priority_group = "Child/Youngster";
    } else if (patients[patientCount].age > 65) {
        patients[patientCount].priority_group = "Elderly";
    } else {
        patients[patientCount].priority_group = "General";
    }

    patientCount++;
}

```

```
q3.sh | q2.cpp | ● q2.c | ● question1 x
}

int main() {
    const int MAX_PATIENTS = 100;
    Patient patients[MAX_PATIENTS];
    int patientCount = 0;
    cout << " moneeb toor , Roll# : 221048" <<endl;
    cout << "Enter the number of patients: ";
    int n;
    cin >> n;

    for (int i = 0; i < n; i++) {
        processPatient(patients, patientCount);
    }

    priority_queue<Patient, vector<Patient>, EmergencyComparator> emergencyQueue;
    priority_queue<Patient, vector<Patient>, PriorityComparator> priorityQueue;
    priority_queue<Patient, vector<Patient>, PriorityComparator> generalQueue;

    for (int i = 0; i < patientCount; i++) {
        if (patients[i].priority_group == "Emergency") {
            emergencyQueue.push(patients[i]);
        } else if (patients[i].priority_group == "Child/Youngster" || patients[i].priority_group == "Elderly") {
            priorityQueue.push(patients[i]);
        } else {
            generalQueue.push(patients[i]);
        }
    }

    int servedPatients = 0;
    while (servedPatients < n) {
        if (!emergencyQueue.empty()) {
            Patient p = emergencyQueue.top();
            emergencyQueue.pop();
            cout << "Serving emergency patient " << p.id << " with estimated time " << p.estimated_time << " minutes.
\n";

            servedPatients++;
        }
        else if (!priorityQueue.empty()) {
            Patient p = priorityQueue.top();
            priorityQueue.pop();
            cout << "Serving priority patient " << p.id << " with estimated time " << p.estimated_time << " minutes.
\n";

            servedPatients++;
        }
        else if (!generalQueue.empty()) {
            Patient p = generalQueue.top();
            generalQueue.pop();
            cout << "Serving general patient " << p.id << " with estimated time " << p.estimated_time << " minutes.
```

```

        else if (!priorityQueue.empty()) {
            Patient p = priorityQueue.top();
            priorityQueue.pop();
            cout << "Serving priority patient " << p.id << " with estimated time " << p.estimated_time << " minutes.
\n";
            servedPatients++;
        }
        else if (!generalQueue.empty()) {
            Patient p = generalQueue.top();
            generalQueue.pop();
            cout << "Serving general patient " << p.id << " with estimated time " << p.estimated_time << " minutes.
\n";
            servedPatients++;
        }
    }
    return 0;
}

```

OUTPUT

```

moneeb-toor@moneeb-toor-VMware-Virtual-Platform:~/Desktop$ g++ -o question1 question1.cpp
moneeb-toor@moneeb-toor-VMware-Virtual-Platform:~/Desktop$ ./question1
moneeb Roll# : 221048
Enter the number of patients: 4
Enter patient ID: 212
Is this an emergency (1 for Yes, 0 for No)? 1
Enter patient's age: 32
Enter estimated consultation time (in minutes): 15
Enter patient ID: 213
Is this an emergency (1 for Yes, 0 for No)? 0
Enter patient's age: 23
Enter estimated consultation time (in minutes): 25
Enter patient ID: 112
Is this an emergency (1 for Yes, 0 for No)? 1
Enter patient's age: 12
Enter estimated consultation time (in minutes): 12
Enter patient ID: 12
Is this an emergency (1 for Yes, 0 for No)? 1
Enter patient's age: 13
Enter estimated consultation time (in minutes): 35
Serving emergency patient 112 with estimated time 12 minutes.
Serving emergency patient 212 with estimated time 15 minutes.
Serving emergency patient 12 with estimated time 35 minutes.
Serving general patient 213 with estimated time 25 minutes.
moneeb-toor@moneeb-toor-VMware-Virtual-Platform:~/Desktop$

```

QUE 2



```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main() {
    printf("221048");
    printf("moneeb toor");
    for (int i = 0; i < 10; i++) {
        if (i % 2 == 0) {
            pid_t pid = fork();

            if (pid == -1) {
                perror("Fork failed");
                exit(1);
            }
            if (pid == 0) {
                printf("Child created for call %d\n", i);
                exit(0);
            }
        } else {
            printf("No child created for call %d\n", i);
        }
    }
    sleep(5);
    return 0;
}
```

OUTPUT

```
moneeb-toor@moneeb-toor-VMware-Virtual-Platform:~/Desktop$ ./q2
221048moneeb toorNo child created for call 1
No child created for call 3
221048moneeb toorChild created for call 0
Child created for call 2
Child created for call 4
No child created for call 5
No child created for call 7
No child created for call 9
Child created for call 6
Child created for call 8
^[moneeb-toor@moneeb-toor-VMware-Virtual-Platform:~/Desktop$
```

Que 3

```
Open ▾  q3.sh
~/Desktop

q3.sh

#!/bin/bash
echo "MONEEB TOOR "
echo "221048"
echo " Enter two number  "
read num1
read num2

echo "Select the operation you want to perform:"
echo "1) Addition"
echo "2) Subtraction"
echo "3) Multiplication"
echo "4) Division"
read choice

case $choice in
  1) echo "Result: $((num1 + num2))" ;;
  2) echo "Result: $((num1 - num2))" ;;
  3) echo "Result: $((num1 * num2))" ;;
  4) if [ $num2 -ne 0 ]; then
      echo "Result: $((num1 / num2))"
    else
      echo "Error: Division by zero is not allowed"
    fi
    ;;
  *) echo "Invalid choice" ;;
esac
```

```
moneeb-toor@moneeb-toor-VMware-Virtual-Platform: ~/Desktop

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

moneeb-toor@moneeb-toor-VMware-Virtual-Platform:~$ q3.sh
q3.sh: command not found
moneeb-toor@moneeb-toor-VMware-Virtual-Platform:~$ touch q3.sh
moneeb-toor@moneeb-toor-VMware-Virtual-Platform:~$ cd desktop
bash: cd: desktop: No such file or directory
moneeb-toor@moneeb-toor-VMware-Virtual-Platform:~$ cd Desktop
moneeb-toor@moneeb-toor-VMware-Virtual-Platform:~/Desktop$ chmod +x q3.sh
moneeb-toor@moneeb-toor-VMware-Virtual-Platform:~/Desktop$ ./q3.sh
MONEEB TOOR
221048
 Enter two number
2
3
Select the operation you want to perform:
1) Addition
2) Subtraction
3) Multiplication
4) Division
1
Result: 5
moneeb-toor@moneeb-toor-VMware-Virtual-Platform:~/Desktop$
```

```
chmod: cannot access 'q3.sh': No such file or directory
moneeb-toor@moneeb-toor-VMware-Virtual-Platform:~$ cd Desktop
moneeb-toor@moneeb-toor-VMware-Virtual-Platform:~/Desktop$ chmod +x q3.sh
moneeb-toor@moneeb-toor-VMware-Virtual-Platform:~/Desktop$ ./q3.sh
MONEEB TOOR
221048
Enter two number
3
3
Select the operation you want to perform:
1) Addition
2) Subtraction
3) Multiplication
4) Division
2
Result: 0
moneeb-toor@moneeb-toor-VMware-Virtual-Platform:~/Desktop$
```