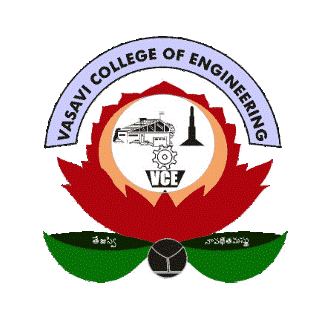
**VASAVI COLLEGE OF ENGINEERING**

**(Autonomous)**



Department of Computer Science Engineering

**MINI PROJECT**

**TITLE**: WAITER’S TOOL

**SUBJECT**: DESIGN AND ALGORITHMS

**GROUP MEMBERS**:

K. RAKSHITH REDDY 1602-17-733-027

B.CHANDRA KIRAN 1602-17-733-005

**CONTENTS**

1. INTRODUCTION

1.1. OVERVIEW

1.2. PURPOSE,SCOPE AND

OBJECTIVE

2. MODULES

2.1. FUNCTION DECLARATIONS

3. SOURCE CODE

3.1. FUNCTION DEFINITIONS

3.2. MAIN FUNCTION

4.OUTPUT SCREENSHOTS

**INTRODUCTION**

**OVERVIEW:**

This application is built based on the waiter’s point of view and is applicable for any given instance and constraints. The algorithm used is job sequencing using deadlines.This app also gives a brief idea of how a restaurant or a small size café carries out its bookings. This program checks all possibilities and gives the best and most efficient outcome. It serves as a helping tool to all the waiter’s to gain the maximum profit(tip) by serving to a selection number of tables in a given time.The profit on each table is measured by taking the number of people on a table as input and deadlines are allocated based in the entry times.

**PURPOSE,SCOPE AND OBJECTIVE:**

The sole purpose and objective of building this application is to implement the algorithm to a real life situation. In this application the algorithm used is job sequencing using deadlines. The entry times and the number of people on each table are taken as inputs.

**MODULES**

**FUNCTION DECLARATIONS:**

1. void jobSequencingWithDeadline(table jobs[], int n);

2. int minValue(int x, int y);

3. Assignment of deadlines(included in main)

4. Assignment of profits(included in main)

**SOURCE CODE:**

**1.FUNCTION DEFINITIONS:**

int minValue(int x, int y) {

if(x < y) return x;

return y;

}

void jobSequencingWithDeadline(table jobs[], int n) {

int i, j, k, maximumtip;

int timeslot[MAX];

int filledTimeSlot = 0;

int dmax = 0;

for(i = 0; i< n; i++) {

if(jobs[i].deadline >dmax) {

dmax = jobs[i].deadline;

}

}

for(i = 1; i<= dmax; i++) {

timeslot[i] = -1;

}

printf("\nMAXIMUM NO OF TABLES HE CAN SERVE IS : %d\n", dmax);

for(i = 1; i<= n; i++) {

k = minValue(dmax, jobs[i - 1].deadline);

while(k >= 1) {

if(timeslot[k] == -1) {

timeslot[k] = i-1;

filledTimeSlot++;

break;

}

k--;

}

if(filledTimeSlot == dmax) {

break;

}

}

printf("\nSEQUENCE TO BE FOLLOWED TO GET MAXIMUM TIP IS : ");

for(i = 1; i<= dmax; i++) {

printf("TABLE %s", jobs[timeslot[i]].id);

if(i<dmax) {

printf(" --> ");

}

}

maximumtip = 0;

for(i = 1; i<= dmax; i++) {

maximumtip+= jobs[timeslot[i]].profit;

}

printf("\n\nMax tip he can receive is: %d\n", maximumtip);

}

**2.MAIN FUNCTION**

#include <stdio.h>

#include<stdlib.h>

#define MAX 100

#define TIME 420

typedef struct table {

char id[5];

int entrytime\_hr;

int entrytime\_min;

int deadline;

int profit;

} table;

void jobSequencingWithDeadline(table jobs[], int n);

int minValue(int x, int y);

int main()

{

int i, j,per=0,k;

int n = 6;

table jobs[6] = {

{"t1", 7, 30, 0, 0},

{"t2", 7, 10, 0, 0},

{"t3", 7, 15, 0, 0},

{"t4", 7, 25, 0, 0},

{"t5", 7, 40, 0, 0},

{"t6", 7, 35, 0, 0},

};

printf("Enter the entry times and number of people on all the tables\n ");

table temp;

for(i=0;i<n;i++)

{

k=0;

printf("\nEnter the number of people on table no ->%d ",i+1);

scanf("%d",&per);

jobs[i].profit=per\*15;

do

{

k++;

if(k>=2)

printf("invalid entry\n");

printf("Enter the entry time of table no ->%d ",i+1);

scanf("%d:%d",&jobs[i].entrytime\_hr,&jobs[i].entrytime\_min);

}while(jobs[i].entrytime\_min>=60||jobs[i].entrytime\_hr<7);

jobs[i].deadline= (((jobs[i].entrytime\_hr\*60)+jobs[i].entrytime\_min)-TIME);

if(jobs[i].deadline>=45)

jobs[i].deadline=4;

else if(jobs[i].deadline<45&&jobs[i].deadline>=25)

jobs[i].deadline=3;

else if(jobs[i].deadline<25&&jobs[i].deadline>=10)

jobs[i].deadline=2;

else

jobs[i].deadline=1;

}

for(i = 1; i< n; i++) {

for(j = 0; j < n - i; j++) {

if(jobs[j+1].profit > jobs[j].profit) {

temp = jobs[j+1];

jobs[j+1] = jobs[j];

jobs[j] = temp;

}

}

}

printf("\n\n\n");

printf("%10s %10s %10s\n", "Job", "Entrytime", "Profit");

for(i = 0; i< n; i++) {

printf("%10s %10d:%d %10i\n", jobs[i].id, jobs[i].entrytime\_hr,jobs[i].entrytime\_min, jobs[i].profit);

}

jobSequencingWithDeadline(jobs, n);

return 0;

}

**OUTPUT SCREENSHOTS:**

