James Kenneth M. Kiunisala
BSCS 1-1

Data Structures & Algorithms
Source Code & Sample Outputs

Machine Problem No. 1

The results from the mayor's race have been reported by each precinct as follows:

Precincts	Candidate A	Candidate B	Candidate C	Candidate D
1	192	48	206	37
2	147	90	312	21
3	186	12	121	38
4	114	21	408	39
5	267	13	382	29

Write a program to do the following:

a. Print out the table with the appropriate headings for the rows and columns.
b. Compute and print the total number of votes received by each candidate and the percent of the total votes cast.
c. If any one candidate received over 50 percent of the total votes, the program should print a message declaring that candidate the winner.
d. If no candidate received 50 percent of the votes, the program should print a message declaring a run-off between the two candidates who received the highest number of votes; two candidates should be identified by their letter names.
e. Run the program once with the preceding data and once with candidate C receiving only 108 votes in precinct 4. *(note: test values are not limited to the preceding data)*

Note: Use Array for this machine problem

Source Code:

```
#include <stdio.h>
#include <conio.h>
#include <windows.h> // for SetConsoleCursorPosition()

void gotoxy (short x, short y); // creating gotoxy() for non-DOS compiler
void layout();

int main() {
        layout();
```

```
int votes[5][4], precintVotes[5], candidateVotes[4], totalVotes;
float voteCastsPer[4];

int i, j, y = 7, x = 33; // x and y for positioning the cursor

for(i = 0; i <= 4; ++i) {
	for(j = 0; j <= 3; ++j) {
		gotoxy(x, y);
		scanf("%d", &votes[i][j]);
		totalVotes += votes[i][j];
		x += 17;
	}
	x = 33; // Reset column to x = 33 (cursor)
	y += 2; // Changing row (cursor placement)
}

// Calculating the Sum per Precint (per Row)
for(i = 0; i <= 4; ++i) {
	int sumRow = 0;
	for(j = 0; j <= 3; ++j) {
		precintVotes[i] = sumRow += votes[i][j];
	}
}

// Output of Votes per Precint
y = 7;
for(i = 0; i <= 4; ++i) {
	gotoxy(102, y);
	printf("%d", precintVotes[i]);
	y += 2;
}

// Calculating the Sum per Candidates
gotoxy(27, 23);
for(i = 0; i <= 3; ++i) {
	int sumCol = 0;
	for(j = 0; j <= 4; ++j) {
		candidateVotes[i] = sumCol += votes[j][i];
	}
}

// Printing Total Votes per Candidate
x = 33;
```

```
for(i = 0; i <= 3; ++i) {
		gotoxy(x, 17);
		printf("%d", candidateVotes[i]);
		x += 17;
}

// Priting Total Votes
gotoxy(102, 17); printf("%d", totalVotes);

// Calculating the Vote Casts Percentage per Candidate
for(i = 0; i <= 3; ++i) {
	voteCastsPer[i] = ((float)candidateVotes[i] / totalVotes) * 100;
}

// Printing the Vote Casts Percentage per Candidate
x = 33;
for(i = 0; i <= 3; ++i) {
	gotoxy(x, 19);
	printf("%.2f", voteCastsPer[i]);
	x += 17;
}

// Priting Total Votes Casts Perdcentage
gotoxy(101, 19);
float totalVoteCastsPer;
for(i = 0; i <= 3; ++i) {
	totalVoteCastsPer += voteCastsPer[i];
}
printf("%.2f", totalVoteCastsPer);

// Declaration of Winner
char candidates[4] = {'A', 'B', 'C', 'D'};
// Checking for the Winner
for(i = 0; i <= 3; ++i) {
	if(voteCastsPer[i] > 50.00) {
		gotoxy(40, 22);
		printf("RESULT: The winner is Candidate %c!", candidates[i]);
		break;
	}
}

// Checking for run-off
if(voteCastsPer[i] < 50) {
	// Finding the Highest 2 values in the voteCastsPer
```

```
                int highest, secondHighest, idxHigh, idxSecHigh; // Index for Highest and Second
Highest

                for(i = 0; i < sizeof(voteCastsPer); ++i) {
                   if(voteCastsPer[i] > highest) {
                      secondHighest = highest;// Making previous highest percentage to Second
Highest
                      idxSecHigh = idxHigh;      // Identifying the new highest vote Casts percentage
                      highest = voteCastsPer[i];
                      idxHigh = i;
                   }
                   else if(voteCastsPer[i] > secondHighest) {
                      secondHighest = voteCastsPer[i];
                      idxSecHigh = i;
                }
                        gotoxy(28, 22);
                        printf("RESULT: There will be a run-off between Candidates %c and %c!",
candidates[idxSecHigh], candidates[idxHigh]);
        }
        }

        getch();
        return(0);
}

void layout() {
        gotoxy(35, 2); printf("\t\tRESULT FOR MAYOR ELECTION");
        gotoxy(10, 3); printf("----------------------------------------------------------------------------------------
---------");
        gotoxy(10, 4); printf("\t\t|");
        gotoxy(30, 4); printf("Candidate \tCandidate    \tCandidate   \tCandidate");
        gotoxy(10, 5); printf("  Precints\t|\t  A\t\t   B\t\t   C\t\t   D\t\t   TOTAL");
        gotoxy(10, 6); printf("----------------------------------------------------------------------------------------
---------");
        gotoxy(10, 7); printf("\t1\t|");
        gotoxy(24, 8); printf("|");
        gotoxy(10, 9); printf("\t2\t|");
        gotoxy(24, 10); printf("|");
        gotoxy(10, 11); printf("\t3\t|");
        gotoxy(24, 12); printf("|");
        gotoxy(10, 13); printf("\t4\t|");
        gotoxy(24, 14); printf("|");
        gotoxy(10, 15); printf("\t5\t|");
```

```
	gotoxy(10, 16); printf("-----------------------------------------------------------------------------------------------");
	gotoxy(10, 17); printf("  TOTAL VOTES\t|");
	gotoxy(10, 18); printf("-----------------------------------------------------------------------------------------------");
	gotoxy(10, 19); printf("  PERCENTAGE\t|");
	gotoxy(10, 20); printf("-----------------------------------------------------------------------------------------------");
}

void gotoxy(short x, short y) {
	COORD pos = {x,y};
	SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), pos);
}
```

Sample Output:

RESULT FOR MAYOR ELECTION

Precincts	Candidate A	Candidate B	Candidate C	Candidate D	TOTAL
1	192	48	206	37	483
2	147	90	312	21	570
3	186	12	121	38	357
4	114	21	408	39	582
5	267	13	382	29	691
TOTAL VOTES	906	184	1429	164	2683
PERCENTAGE	33.77	6.86	53.26	6.11	100.00

RESULT: The winner is Candidate C!

Output 1: *Candidate C gathered 408 votes from Precinct 4*

RESULT FOR MAYOR ELECTION

Precincts	Candidate A	Candidate B	Candidate C	Candidate D	TOTAL
1	192	48	206	37	483
2	147	90	312	21	570
3	186	12	121	38	357
4	114	21	108	39	282
5	267	13	382	29	691
TOTAL VOTES	906	184	1129	164	2383
PERCENTAGE	38.02	7.72	47.38	6.88	100.00

RESULT: There will be a run-off between Candidates A and C!

Output 2: *Candidate C gathered 108 votes from Precinct 4*