**LAB-13**

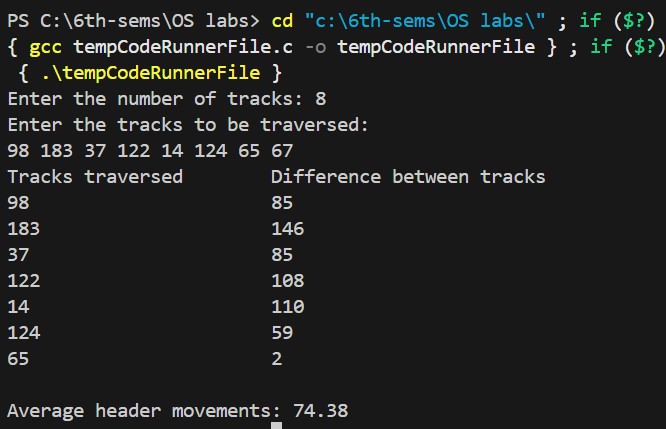
**Exercise:**

1) Implement the above code and paste the screen shot of the output. **a) FCFS**

**PROGRAM:**

|  |
| --- |
| #include <stdio.h>  #include <conio.h>  int main() { int t[20], tohm[20], n, i, tot = 0; float avhm;  // clrscr(); printf("Enter the number of tracks: "); scanf("%d", &n);  printf("Enter the tracks to be traversed:\n"); for (i = 1; i <= n; i++) { scanf("%d", &t[i]);  } for (i = 1; i < n; i++) { tohm[i] = t[i + 1] - t[i]; if (tohm[i] < 0) tohm[i] \*= -1;  } for (i = 1; i < n; i++) { tot += tohm[i];  }  avhm = (float)tot / n;    printf("Tracks traversed\tDifference between tracks\n"); for (i = 1; i < n; i++) { printf("%d\t\t\t%d\n", t[i], tohm[i]);  } printf("\nAverage header movements: %.2f", avhm); getch(); return 0;  } |

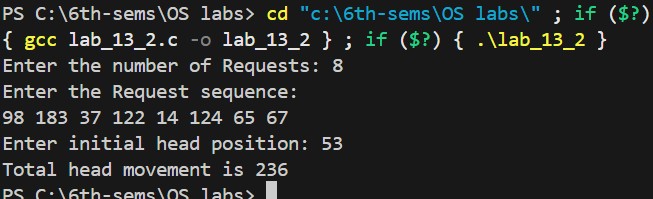
**OUTPUT:**



1. **SSTF**

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  int main() { int RQ[100], n, i, initial, count = 0, TotalHeadMovement = 0;  printf("Enter the number of Requests: "); scanf("%d", &n);    printf("Enter the Request sequence:\n"); for (i = 0; i < n; i++) { scanf("%d", &RQ[i]);  }  printf("Enter initial head position: "); scanf("%d", &initial);  while (count != n) {  int min = 1000, d, index;  for (i = 0; i < n; i++) { d = abs(RQ[i] - initial); if (RQ[i] != 1000 && d < min) { min = d; index = i;  } |
| }    TotalHeadMovement += min; initial = RQ[index];  RQ[index] = 1000; // Mark as visited count++;  } printf("Total head movement is %d\n", TotalHeadMovement); return 0;  } |

**OUTPUT:**



1. **SCAN**

|  |
| --- |
| #include <stdio.h> #include <conio.h> int main() { int t[20], atr[20], d[20], h, i, j, n, temp, k, p = 0, sum = 0;  clrscr(); printf("Enter the number of tracks to be traversed: "); scanf("%d", &n);  printf("Enter the position of head: "); scanf("%d", &h);    t[0] = 0; // Start from 0  t[1] = h; // Current head position    printf("Enter the track numbers:\n"); for (i = 2; i < n + 2; i++) { scanf("%d", &t[i]);  }  // Sort the tracks for (i = 0; i < n + 2; i++) { for (j = 0; j < (n + 2) - i - 1; j++) { if (t[j] > t[j + 1]) { temp = t[j]; t[j] = t[j + 1]; |
| t[j + 1] = temp;  }  }  }  // Find head index for (i = 0; i < n + 2; i++) { if (t[i] == h) { j = i; k = i; break;  }  }  // Traverse left of head while (t[j] != 0) { atr[p++] = t[j--];  } atr[p++] = t[j]; // add 0  // Traverse right of head for (i = k + 1; i < n + 2; i++) { atr[p++] = t[i];  }    // Calculate distances for (i = 0; i < n + 1; i++) { if (atr[i] > atr[i + 1]) d[i] = atr[i] - atr[i + 1]; else d[i] = atr[i + 1] - atr[i]; sum += d[i];  } printf("\nAverage header movements: %.2f\n", (float)sum / n); getch(); return 0;  } |

**OUTPUT:**

