```
avg_tat = (float) total At \n;
printf("\n\n average waiting time = %.2f", avg-wt);
printf("\n Average turnaround time = %.2f \n", avg_tat);

}
```

Name - Bhavna   Course - BSCIT   Std id - 20052005

Q1.
```
 # include <assert.h>
  #include <ctype.h>
  # include <limits.h>
 # include <stdbool.h>
 # include <math.h>
 # include <stddef.h>
 # include <stdio.h>
 # include <stdlib.h>

char* readline();
char *ltrim(char*);
char *rtrim(char*);
char **split_string(char*);
int parse_int(char*);

#
 int main()
   {
   file* fptr = fopen(getenv("output_path"), "w");
   int n = parse_int(ltrim(rtrim(readline())));
```

Name - Bhavna   Course - BSCIT   Std Id - 20052005

Q1.

```
# include <ass ert. h>
#include <ctype . h>
# include <limits. h>
# include < std bool. h>
# include < math. h >
# include < stddef. h>
# include < Stdio . h>
# include < Std lib . h>

char* red line ();
char * ltrim (char *);
Char *rtrim (char*);
Char **split_string(char*);
int parse_int (char*);

int main ()
 {
 file *f ptr = fopen(getenv ("output_path"), "w");
 int n = parse_int (ltrim (rtrim(read line ())));
int** customers = malloc (n* size of (int*));
 for (int i=0; i<n; i++){
   *(customers + i) = malloc (2*sizeof (int )));
   Char** customers_temp= split_string(rtrim (readline()))

 for(int j=0; j<2; j++) {
   int customers_item = parse_int(*(customers_item_temp+
       j));
   ((customers+ i) +j) =customers_item;
      }
  }
 int result = minimumAverage (n, 2, customers);
 fprintf (fptr, "%d \n", result);
   fclose (fptr);
   return 0;
 }
Char *readline () {
  size_t alloc_length =1024;
  size_t data length = 0;

}
```

```
Char *readline() {

~~Str~~

Char *data= malloc(alloc_length);

while (true) {

Char * cursor = data + data_length;
Char * line = fgets(cursor_ alloc_length - data_length,
   stdin);

if(!line) {
break;
}

data_length += strlen(cursor);
if (data_length < alloc_length-1 || data[data_length-1]=='\n')
    {
    break;
    }
```

Name - Bhavna
Course - BSc IT
Std Id - 20052005

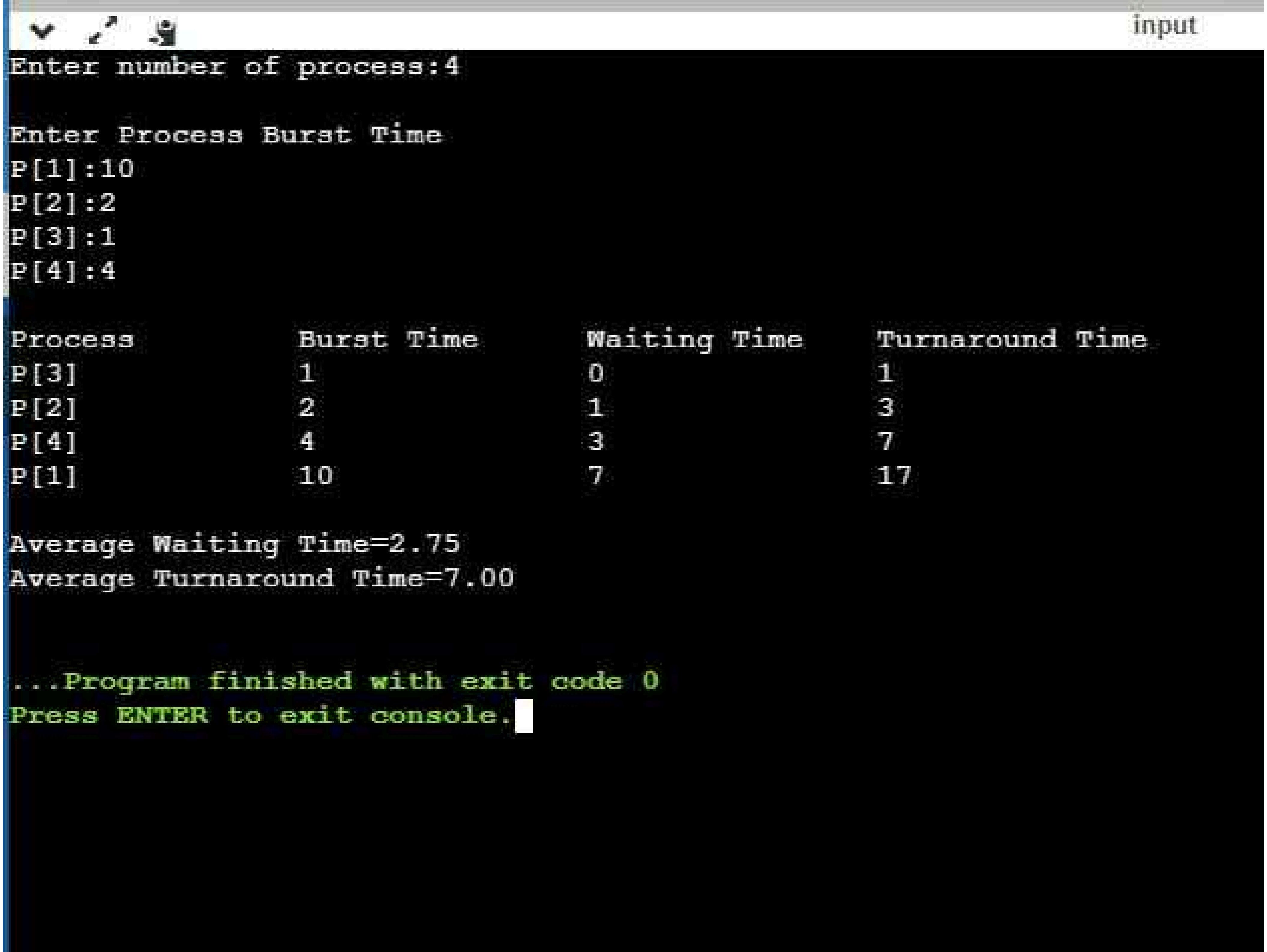
Q 2

```
#include <stdio.h>
int main ()
{
 int bt [20] , p [20], wt[20], tat [20] , i, j, n, total = 0, pos, tem
 float avg_wt, avg_tat;
 printf("Enter no. of processes :");
 Scanf(" %d", &n);
 printf("\n Enter Burst Time of process \n ");
 for(i=0; i<n; i++)
  {
    printf("P[%d]:", i+1);
    Scanf(" %d", &bt[i]);
    P[i] = i+1;

 for (i=0; i<n; i++)
  {
   pos = i;
   ~~for(j=1;~~
   for(j = i+1; j<n; j++)
    {
      if (bt[j]<bt [pos])
        pos = j;
    }
    temp = P[i] ;
    P[i] = P[Pos] ;
     P[Pos] = temp ;
  }
   wt[0] = 0;
  for (i=1; i<n; i++)
   {
     wt[i] =0;
    for(j=0; j<i ; j++)
      wt[i]+= bt[j];
     total + = wt[i] ;
   }
  avg_wt = (float) total /n ;
     total = 0;
printf ("\n process \t\t burst time \t waiting time
    \t turnaround -time");
for(i=0; i<n; i++)
   {
      tat[i] = bt[i] + wt[i];
     total += tat [i];
     printf ("\nP[%d] \t\t %d \t\t %d \t\t %d ", P[i],
      bt[i], wt[i], tat[i]);
   }
```

```
avg_tat= (float) total \n ;
printf("\n\n average waiting time = %.2f", avg_wt);
printf("\n Average turnaround time = %.2f \n", avg_tat);

}
```



```
Enter number of process:4

Enter Process Burst Time
P[1]:10
P[2]:2
P[3]:1
P[4]:4

Process         Burst Time      Waiting Time    Turnaround Time
P[3]            1               0               1
P[2]            2               1               3
P[4]            4               3               7
P[1]            10              7               17

Average Waiting Time=2.75
Average Turnaround Time=7.00


...Program finished with exit code 0
Press ENTER to exit console.
```