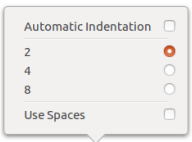
1. PART

1.part.c (~/Desktop) - gedit

```
/*
**************************************************************FATİH****OĞUZ**********************************************************
**************************************************************151044025*************************************************************
*/


#include <stdio.h>                                      /*  printf-scanf fonksiyonları kütüphanesi                                  */
#include <stdlib.h>                                     /*  srand fonksiyonu kütüphanesi                                            */
#include <time.h>                                       /*  srand fonksiyonun zaman ile değişen rastgele sayı üretmesi için kütüphane */

char letter_grade(int grade);

int main()
{
  char letter;

  int flag = 0;                                         /*  student count'un 3-5 arası girdiği zaman döngüden çıkmak için flag      */
  int menu=1;
  int student_count;
  int count=1,
  count_a=0,count_b=0,count_c=0,count_d=0,count_f=0;
  int successful_index ,unsuccessful_index,
  most_successful_student=0, most_unsuccessful_student=100;
  int grade;

  double sum = 0.0;


  while(flag != 1)
  {
    printf("Enter student count:");
    scanf("%d",&student_count);                                     /*  öğrenci sayısını alır                                   */

    if(student_count >=3 && student_count<=50)                      /*  öğrenci sayısı kontrolü yapılır                         */
    {
      flag = 1;                                                     /*  kontrol tutuyorsa flag = 1 yapılır ve öğrenci sayısı alma döngüsü tekrarlanmaz  */
      srand(40);                                                    /*  rastgele not üretmek için fonksiyon                     */


          while(count <= student_count)                             /*  öğrenci sayısınca rastgele sayı basma kontrolü          */
         {

                    grade=rand()%101;                               /*  0 - 100 arasında rastgele sayı bastırma                 */

                    if(most_successful_student < grade)
                    {
                      most_successful_student = grade;              /*  most_successful_student 'ın puanı ve indexini belirleme */
                      successful_index = count;
                    }

                    if(most_unsuccessful_student > grade)           /*  most_unsuccessful_student 'ın puanı ve indexini belirleme
                     {
                      most_unsuccessful_student = grade;
                       unsuccessful_index = count;
                     }

                    /*  rastgele üretilen puanların harf notunu belirleyip her harf notun count ile adetini belirleme
                    if(grade >= 90 && grade <= 100)
```



C Tab Width: 2 Ln 1, Col 1 INS

1.part.c (~/Desktop) - gedit

```
                      most_unsuccessful_student = grade;
                       unsuccessful_index = count;
                     }

                    /*  rastgele üretilen puanların harf notunu belirleyip her harf notun count ile adetini belirleme  */
                    if(grade >= 90 && grade <= 100)
                    {
                      count_a++;
                    }
                    else if(grade >= 80 && grade <= 89)
                    {
                      count_b++;
                    }
                    else if(grade >= 70 && grade <= 79)
                    {
                      count_c++;
                    }
                    else if(grade >= 60 && grade <= 69)
                    {
                      count_d++;
                    }
                    else if(grade >= 0 && grade <= 59)
                    {
                      count_f++;
                    }


                    sum = sum + grade;                          /*  avarage notunu belirlemek için her bir notun toplamını bulma  */
                    printf("%d ",grade);                        /*  puanları printle                                              */
                    count++;                                    /*  count arttılıp while de student_countrolü yap                 */
              }


        }


  else
  {
    printf("Not in Range!!!\n");                                /*  student_count 3 - 5 arasında değilse                          */
  }


}

/*  ekrana menüyü bastır                                                                                                  */

printf("\nStudent Score Calculator Menu for %d Student",student_count);
printf("\n1)Most Successful Student\n");
printf("2)Most Unsuccessful Student\n");
printf("3)Letter Grade Statistics\n");
printf("4)Calculate Avarage\n");
printf("5)Show all Data\n");


while(menu != -1)                                               /*  menu seceneği olarak -1 girerse programı kapat          */
{
  scanf("%d",&menu);
  printf("\t\t\t\t\tMake Selection:%d\n",menu);
```

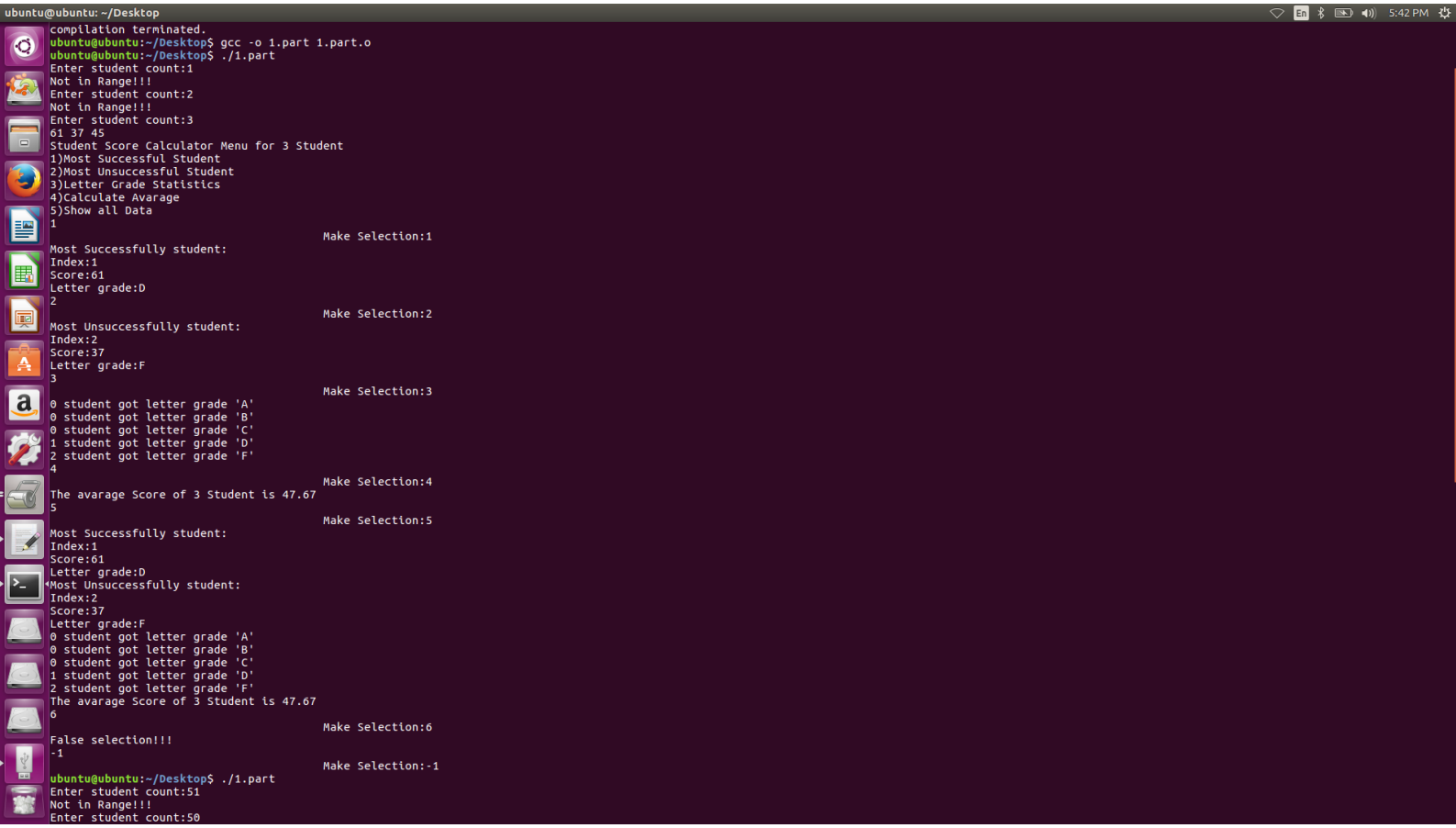
C Tab Width: 2 Ln 1, Col 1 INS

```c
while(menu != -1)                                   /*  menu seceneği olarak -1 girerse programı kapat                  */
{
  scanf("%d",&menu);
  printf("\t\t\t\t\tMake Selection:%d\n",menu);


  switch(menu)
  {

    case -1:                                        /*  -1  girildiğinde false seletion demeden çıkması için break     */
      break;
    case 1:
      letter=letter_grade(most_successful_student);
      printf("Most Successfully student:\nIndex:%d\nScore:%d\nLetter grade:%c\n",successful_index,most_successful_student,letter);

      break;
    case 2:
      letter=letter_grade(most_unsuccessful_student);
      printf("Most Unsuccessfully student:\nIndex:%d\nScore:%d\nLetter grade:%c\n",unsuccessful_index,most_unsuccessful_student,letter);
      break;
    case 3:
      printf("%d student got letter grade 'A'\n",count_a);
      printf("%d student got letter grade 'B'\n",count_b);
      printf("%d student got letter grade 'C'\n",count_c);
      printf("%d student got letter grade 'D'\n",count_d);
      printf("%d student got letter grade 'F'\n",count_f);

      break;
    case 4:
      printf("The avarage Score of %.d Student is %.2f\n",student_count,(sum / (double)student_count));
      break;
    case 5:
      letter=letter_grade(most_successful_student);
      printf("Most Successfully student:\nIndex:%d\nScore:%d\nLetter grade:%c\n",successful_index,most_successful_student,letter);
      letter=letter_grade(most_unsuccessful_student);
      printf("Most Unsuccessfully student:\nIndex:%d\nScore:%d\nLetter grade:%c\n",unsuccessful_index,most_unsuccessful_student,letter);
      printf("%d student got letter grade 'A'\n",count_a);
      printf("%d student got letter grade 'B'\n",count_b);
      printf("%d student got letter grade 'C'\n",count_c);
      printf("%d student got letter grade 'D'\n",count_d);
      printf("%d student got letter grade 'F'\n",count_f);
      printf("The avarage Score of %d Student is %.2f\n",student_count,(sum / (double)student_count));
        break;
    default:
      printf("False selection!!!\n");
      break;

  }


 }
}
```

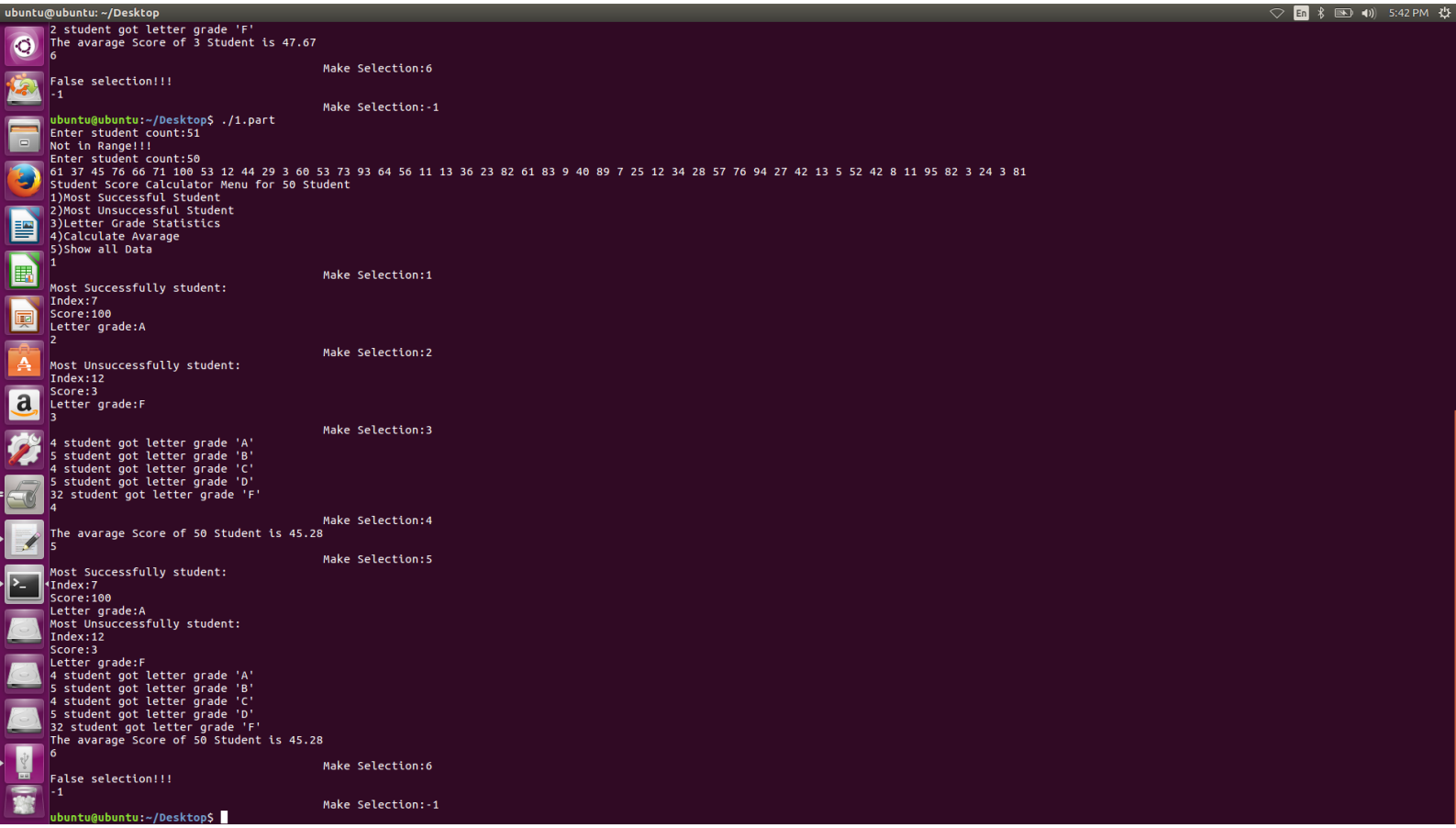


```c
      break;
    case 4:
      printf("The avarage Score of %.d Student is %.2f\n",student_count,(sum / (double)student_count));
      break;
    case 5:
      letter=letter_grade(most_successful_student);
      printf("Most Successfully student:\nIndex:%d\nScore:%d\nLetter grade:%c\n",successful_index,most_successful_student,letter);
      letter=letter_grade(most_unsuccessful_student);
      printf("Most Unsuccessfully student:\nIndex:%d\nScore:%d\nLetter grade:%c\n",unsuccessful_index,most_unsuccessful_student,letter);
      printf("%d student got letter grade 'A'\n",count_a);
      printf("%d student got letter grade 'B'\n",count_b);
      printf("%d student got letter grade 'C'\n",count_c);
      printf("%d student got letter grade 'D'\n",count_d);
      printf("%d student got letter grade 'F'\n",count_f);
      printf("The avarage Score of %d Student is %.2f\n",student_count,(sum / (double)student_count));
        break;
    default:
      printf("False selection!!!\n");
      break;

  }


 }
}

   /*  most_successful_student ve most_unsuccessful_student 'ın harf notunu belirlemek için fonksiyon                 */
char letter_grade(int grade)
{
 char letter;

        if(grade >= 90 && grade <= 100)
        {
          letter = 'A';
        }
        else if(grade >= 80 && grade <= 89)
        {
          letter = 'B';
        }
        else if(grade >= 70 && grade <= 79)
        {
          letter = 'C';
        }
        else if(grade >= 60 && grade <= 69)
        {
          letter = 'D';
        }
        else if(grade >= 0 && grade <= 59)
        {
          letter = 'F';
        }

 return(letter);
}
```

1.PART SONUÇLARI



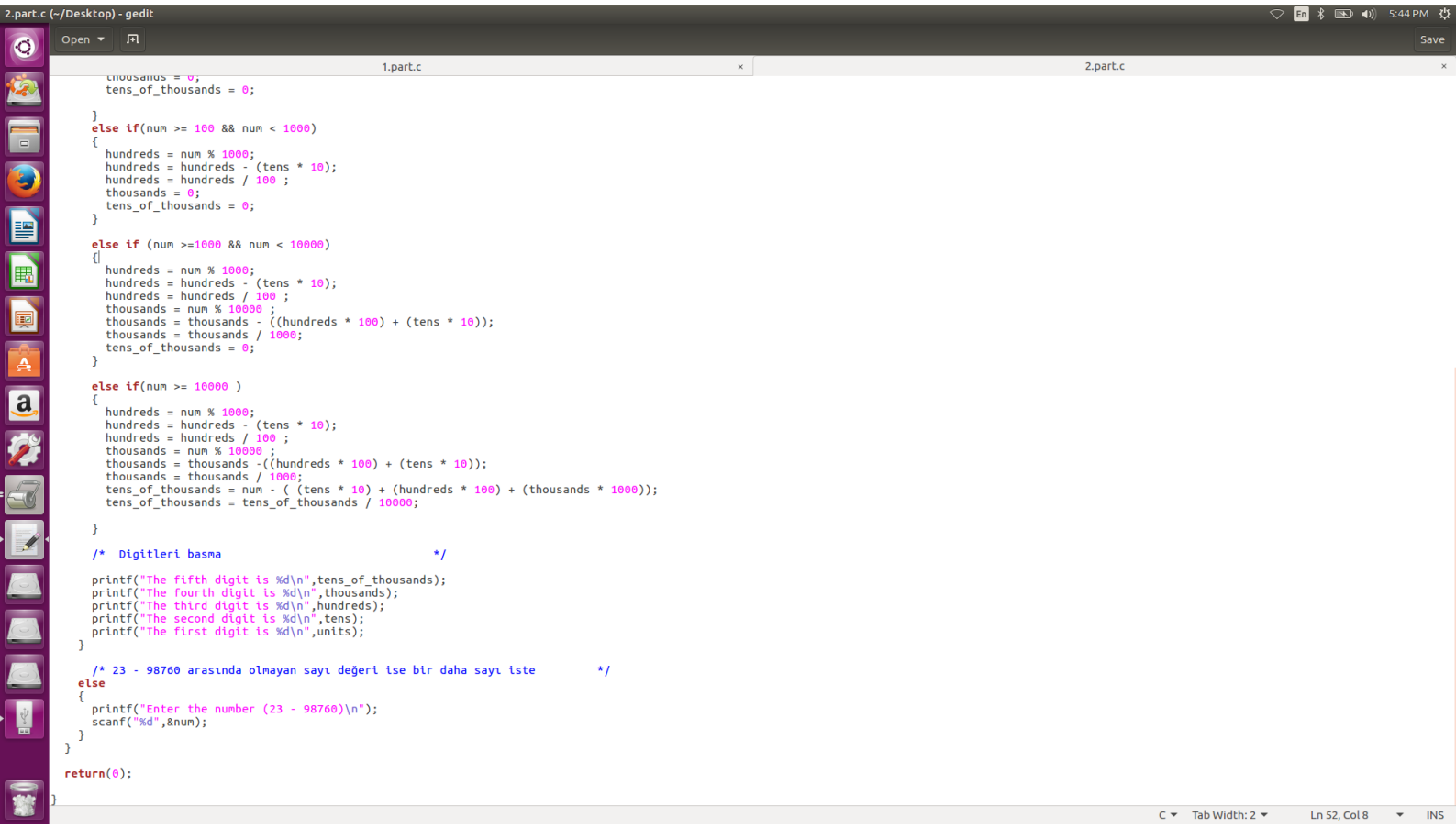
```
ubuntu@ubuntu: ~/Desktop                                                    5:42 PM
compilation terminated.
ubuntu@ubuntu:~/Desktop$ gcc -o 1.part 1.part.o
ubuntu@ubuntu:~/Desktop$ ./1.part
Enter student count:1
Not in Range!!!
Enter student count:2
Not in Range!!!
Enter student count:3
61 37 45
Student Score Calculator Menu for 3 Student
1)Most Successful Student
2)Most Unsuccessful Student
3)Letter Grade Statistics
4)Calculate Avarage
5)Show all Data
1
                                        Make Selection:1
Most Successfully student:
Index:1
Score:61
Letter grade:D
2
                                        Make Selection:2
Most Unsuccessfully student:
Index:2
Score:37
Letter grade:F
3
                                        Make Selection:3
0 student got letter grade 'A'
0 student got letter grade 'B'
0 student got letter grade 'C'
1 student got letter grade 'D'
2 student got letter grade 'F'
4
                                        Make Selection:4
The avarage Score of 3 Student is 47.67
5
                                        Make Selection:5
Most Successfully student:
Index:1
Score:61
Letter grade:D
Most Unsuccessfully student:
Index:2
Score:37
Letter grade:F
0 student got letter grade 'A'
0 student got letter grade 'B'
0 student got letter grade 'C'
1 student got letter grade 'D'
2 student got letter grade 'F'
The avarage Score of 3 Student is 47.67
6
                                        Make Selection:6
False selection!!!
-1
                                        Make Selection:-1
ubuntu@ubuntu:~/Desktop$ ./1.part
Enter student count:51
Not in Range!!!
Enter student count:50
```



```
ubuntu@ubuntu: ~/Desktop                                                    5:42 PM
2 student got letter grade 'F'
The avarage Score of 3 Student is 47.67
6
                                        Make Selection:6
False selection!!!
-1
                                        Make Selection:-1
ubuntu@ubuntu:~/Desktop$ ./1.part
Enter student count:51
Not in Range!!!
Enter student count:50
61 37 45 76 66 71 100 53 12 44 29 3 60 53 73 93 64 56 11 13 36 23 82 61 83 9 40 89 7 25 12 34 28 57 76 94 27 42 13 5 52 42 8 11 95 82 3 24 3 81
Student Score Calculator Menu for 50 Student
1)Most Successful Student
2)Most Unsuccessful Student
3)Letter Grade Statistics
4)Calculate Avarage
5)Show all Data
1
                                        Make Selection:1
Most Successfully student:
Index:7
Score:100
Letter grade:A
2
                                        Make Selection:2
Most Unsuccessfully student:
Index:12
Score:3
Letter grade:F
3
                                        Make Selection:3
4 student got letter grade 'A'
5 student got letter grade 'B'
4 student got letter grade 'C'
5 student got letter grade 'D'
32 student got letter grade 'F'
4
                                        Make Selection:4
The avarage Score of 50 Student is 45.28
5
                                        Make Selection:5
Most Successfully student:
Index:7
Score:100
Letter grade:A
Most Unsuccessfully student:
Index:12
Score:3
Letter grade:F
4 student got letter grade 'A'
5 student got letter grade 'B'
4 student got letter grade 'C'
5 student got letter grade 'D'
32 student got letter grade 'F'
The avarage Score of 50 Student is 45.28
6
                                        Make Selection:6
False selection!!!
-1
                                        Make Selection:-1
ubuntu@ubuntu:~/Desktop$
```

2.PART

```
/*****************************************************************FATİH***OĞUZ*****************************************************************************************************
*****************************************************************151044025*****************************************************************************************************
*/

#include <stdio.h>                                          /*  printf , scanf kütüphanesi                                  */
int main()
{

  int num;
  int units ,tens,hundreds ,thousands, tens_of_thousands;
  int flag = 0;

  printf("Enter the number (23 - 98760)\n");
  scanf("%d",&num);


  while( flag == 0 )
  {
    if( num >= 23 && num <= 98760)                          /*   23 - 98760 arasındaki sayıların basamaklarını ayırmak için koşul  */
    {
      flag = 1;                                             /*  bir daha sayı almaması için flag = 1 yaptık                  */

        /* junk junk basamakları almasın diye her bir basamak grubunda grupta olmayan basamak değerleri 0'a eşitledik          */
        /*  num değerini istenen aralıktan bir üs basamak değerine mod al. (kendi basamak degerine kadar olan kısım sıfırlandı)  */
        /*  istenilen basamak değernden küçük olanları değerlerince çıkar (sayının sağındaki kısımda sıfırlandı)               */
        /*  kendi değerine böl ve digit bul.                                                                                    */

      units = num % 10;
      num = num - units;
      tens = num % 100;
      tens = tens / 10;
      hundreds = 0;
      thousands = 0;
      tens_of_thousands = 0;
      if(num >= 23 && num < 100)
      {
        hundreds = 0;
        thousands = 0;
        tens_of_thousands = 0;

      }
      else if(num >= 100 && num < 1000)
      {
        hundreds = num % 1000;
        hundreds = hundreds - (tens * 10);
        hundreds = hundreds / 100 ;
        thousands = 0;
        tens_of_thousands = 0;
      }

      else if (num >=1000 && num < 10000)
      {
        hundreds = num % 1000;
        hundreds = hundreds - (tens * 10);
        hundreds = hundreds / 100 ;
        thousands = num % 10000 ;
        thousands = thousands - ((hundreds * 100) + (tens * 10));
```



```
        tens_of_thousands = 0;

      }
      else if(num >= 100 && num < 1000)
      {
        hundreds = num % 1000;
        hundreds = hundreds - (tens * 10);
        hundreds = hundreds / 100 ;
        thousands = 0;
        tens_of_thousands = 0;
      }

      else if (num >=1000 && num < 10000)
      {
        hundreds = num % 1000;
        hundreds = hundreds - (tens * 10);
        hundreds = hundreds / 100 ;
        thousands = num % 10000 ;
        thousands = thousands - ((hundreds * 100) + (tens * 10));
        thousands = thousands / 1000;
        tens_of_thousands = 0;
      }

      else if(num >= 10000 )
      {
        hundreds = num % 1000;
        hundreds = hundreds - (tens * 10);
        hundreds = hundreds / 100 ;
        thousands = num % 10000 ;
        thousands = thousands -((hundreds * 100) + (tens * 10));
        thousands = thousands / 1000;
        tens_of_thousands = num - ( (tens * 10) + (hundreds * 100) + (thousands * 1000));
        tens_of_thousands = tens_of_thousands / 10000;

      }

      /*  Digitleri basma                             */

      printf("The fifth digit is %d\n",tens_of_thousands);
      printf("The fourth digit is %d\n",thousands);
      printf("The third digit is %d\n",hundreds);
      printf("The second digit is %d\n",tens);
      printf("The first digit is %d\n",units);
    }

      /* 23 - 98760 arasında olmayan sayı değeri ise bir daha sayı iste        */
    else
    {
      printf("Enter the number (23 - 98760)\n");
      scanf("%d",&num);
    }
  }

  return(0);
}
```

2.PART SONUÇLARI