|  |  |
| --- | --- |
| **NAME:** | VIJESH HINGU |
| **UID:** | 2021300042 |
| **BATCH** | C |
| **SUBJECT** | DAA |
| **EXPERIMENT NO :** | 1B |
| **DATE OF PERFORMANCE** | 06-02-2023 |
| **DATE OF SUBMISSION** | 13-02-2023 |
| **AIM:** | To find the running time of insertion sort and selection sort. |
| **ALGORITHM** | Insertion sort –   1. Access text file from data. 2. For i=0 to n : 3. key=arr[i]; 4. while j>=0 and arr[j]>key : arr[j+1]=arr[j] and j=j-1. 5. arr[j+1]=key. 6. Repeat steps 2 to 5 until i reaches n.   Selection sort –   1. Access text file from data. 2. For i=0 to n-1 and for j=i to n-1 3. If arr[j]<arr[i] and arr[j]<arr[mini] then mini=j. 4. Swap arr[mini] and arr[i]. 5. Repeat steps 3 and 4. |
| **PROGRAM** | #include<stdio.h>  #include<stdlib.h>  #include<time.h>  void insertionsort(int n)  {  int i,key,j;  FILE \*fp;  fp=fopen("daaexp1bdata.txt","r");  int arr[n],arrsort[n];  for(i=0;i<n;i++)  {  fscanf(fp,"%d",&arr[i]);  }  for(i=1;i<n;i++)  {  key=arr[i];  j=i-1;  while(j>=0&&arr[j]>key)  {  arr[j+1]=arr[j];  j=j-1;  }  arr[j+1]=key;  }  fclose(fp);  }  void selectionsort(int n)  {  int i,j;  FILE \*fp;  fp=fopen("daaexp1bdata.txt","r");  int arr[n];  for(int i=0;i<n;i++)  {  fscanf(fp,"%d",&arr[i]);  }  for(i=0;i<n;i++)  {  int temp,mini=i;  for(j=i;j<n;j++)  {  if(arr[j]<arr[i]&&arr[j]<arr[mini])  mini=j;  }  temp=arr[mini];  arr[mini]=arr[i];  arr[i]=temp;  }  fclose(fp);  }  void main()  {  int i;  clock\_t start,end;  FILE \*fp;  fp=fopen("daaexp1bdata.txt","w");  for(int i=0;i<100000;i++)  {  int num=(rand()%100000);  fprintf(fp," %d",num);  }  fclose(fp);  FILE \*fp2;  fp2=fopen("daaexp1bexcel.xlsx","w");  for(i=100;i<=100000;i=i+100)  {  start=clock();  insertionsort(i);  end=clock();  fprintf(fp2," %lf",(double)(end-start)/CLOCKS\_PER\_SEC);  start=clock();  selectionsort(i);  end=clock();  fprintf(fp2," %lf",(double)(end-start)/CLOCKS\_PER\_SEC);  fprintf(fp2,"\n");  }  fclose(fp2);  } |
| **RESULT ( SNAPSHOT):**   |  |  |  | | --- | --- | --- | | n | Insertion sort | Selection sort | | 10000 | 0.07 | 0.27 | | 20000 | 0.27 | 1.08 | | 30000 | 0.63 | 2.43 | | 40000 | 1.09 | 4.3 | | 50000 | 1.71 | 6.76 | | 60000 | 2.53 | 9.76 | | 70000 | 3.37 | 13.25 | | 80000 | 4.4 | 17.25 | | 90000 | 5.54 | 21.91 | | 100000 | 6.87 | 27.07 | | |

**CONCLUSION :**

With the help of this experiment, I was able to understand and implement insertion sort and selection sort. I was able to differentiate between the runtimes of bot the algorithms for different number of input values.