sort.c

```c
#include<stdio.h>
#include <stdlib.h>
#include <time.h>
clock_t start, end;
double cpu_time;

void efficient_bubble(int arr[], int n)
{
    int i,j,temp,flag;
    for(i=0;i<n-1;i++)
    {
        flag=0;
        for(j=0;j<n-i-1;j++)
        {
            if(arr[j]>arr[j+1])
            {
                flag=1;
                temp=arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=temp;
            }
        }
        if(flag==0)
        {
            break;
        }
    }
    printf("\nSorted array is:\n");
    for(i=0;i<n;i++)
    {
        printf("%d ",arr[i]);
    }
}
void bubble(int arr[], int n)
{
    int i,j,temp;
    for(i=0;i<n-1;i++)
    {
```

sort.c

```c
{
    int i,j,temp;
    for(i=0;i<n-1;i++)
    {
        for(j=0;j<n-i-1;j++)
        {
            if(arr[j]>arr[j+1])
            {
                temp=arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=temp;
            }
        }
    }
    printf("\n Sorted array is:\n");
    for(i=0;i<n;i++)
    {
        printf("%d ",arr[i]);
    }
}

int main()
{
    int i,n,c,d,k,flag=1,choice,arr[10000];
    srand(time(0));
    while(flag==1)
    {
        printf("\n1:Efficient Bubble Sort\n2:Bubble Sort\n3:Exit\n");
        printf("Enter your choice\n");
        scanf("%d", &choice);
        switch(choice)
        {
            case 1:
            printf("Enter the number of elements in array\n");
            scanf("%d", &n);
            printf("Elements of the array are:\n");
            for (i= 0; i<n; i++)
            {
```

sort.c

```c
            printf("Enter the number of elements in array\n");
            scanf("%d", &n);
            printf("Elements of the array are:\n");
            for (i= 0; i<n; i++)
            {
               arr[i]=rand()%100;
               printf("%d ",arr[i]);
            }
            start = clock();
            efficient_bubble(arr,n);
            end = clock();
            cpu_time = (double)(end - start) / CLOCKS_PER_SEC;
            printf("\nExecution time for effiecient bubble sort = %f ms\n", cpu_time*1000);
            break;
            case 2:
            printf("Enter the number of elements in array\n");
            scanf("%d", &n);
            printf("Elements of the array are:\n");
            for (i= 0; i<n; i++)
            {
               arr[i]=rand()%100;
               printf("%d ",arr[i]);
            }
            start = clock();
            bubble(arr,n);
            end = clock();
            cpu_time = (double)(end - start) / CLOCKS_PER_SEC;
            printf("\nExecution time for Bubble Sort = %f ms\n", cpu_time*1000);
            break;
            default:flag=0;
        }
    }
    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: xyz

```
PS C:\Users\muska\OneDrive\Desktop\C programs> gcc sort.c -o xyz
PS C:\Users\muska\OneDrive\Desktop\C programs> .\xyz.exe

1:Efficient Bubble Sort
2:Bubble Sort
3:Exit
Enter your choice
1
Enter the number of elements in array
100
Elements of the array are:
36 38 9 2 55 14 0 79 9 0 62 98 97 16 91 2 84 6 38 43 31 39 76 86 13 25 30 45 86 40 81 22 6 2 57 15 60 24 40 78 23 1 5 14 82 39 61 61 53 51 4 11 99 26 91 56 3 41 97 25 72 72 43 73 64 87 73 29 78 57 89 2
4 59 22 98 1 66 86 69 90 63 54 6 74 90 3 19 16 55 88 53 50 85 49 36 45 19 98 99 42
Sorted array is:
0 0 1 1 2 2 2 3 3 4 5 6 6 6 9 9 11 13 14 14 15 16 16 19 19 22 22 23 24 24 25 25 26 29 30 31 36 36 38 38 39 39 40 40 41 42 43 43 45 45 49 50 51 53 53 54 55 55 56 57 57 59 60 61 61 62 63 64 66 69 72 72 7
3 73 74 76 78 78 79 81 82 84 85 86 86 86 87 88 89 90 90 91 91 97 97 98 98 98 99 99
Execution time for effiecient bubble sort = 16.000000 ms

1:Efficient Bubble Sort
2:Bubble Sort
3:Exit
Enter your choice
2
Enter the number of elements in array
100
Elements of the array are:
91 28 58 34 53 42 19 8 58 98 28 43 42 67 52 31 94 58 63 88 6 44 94 26 90 26 10 45 79 98 41 57 33 26 40 58 15 2 78 4 18 87 60 9 81 28 44 29 88 84 36 87 32 42 9 79 43 72 95 75 92 40 34 66 57 14 37 45 19
29 47 68 58 68 67 41 82 4 80 2 4 13 97 29 77 40 81 47 48 91 30 74 6 99 39 8 71 29 31 80
 Sorted array is:
2 2 4 4 4 6 6 8 8 9 9 10 13 14 15 18 19 19 26 26 26 28 28 28 29 29 29 29 30 31 31 32 33 34 34 36 37 39 40 40 40 41 41 42 42 42 43 43 44 44 45 45 47 47 48 52 53 57 57 58 58 58 58 60 63 66 67 67 68 68
 71 72 74 75 77 78 79 79 80 80 81 81 82 84 87 87 88 88 90 91 91 92 94 94 95 97 98 98 99
Execution time for Bubble Sort = 18.000000 ms

1:Efficient Bubble Sort
2:Bubble Sort
3:Exit
Enter your choice
```