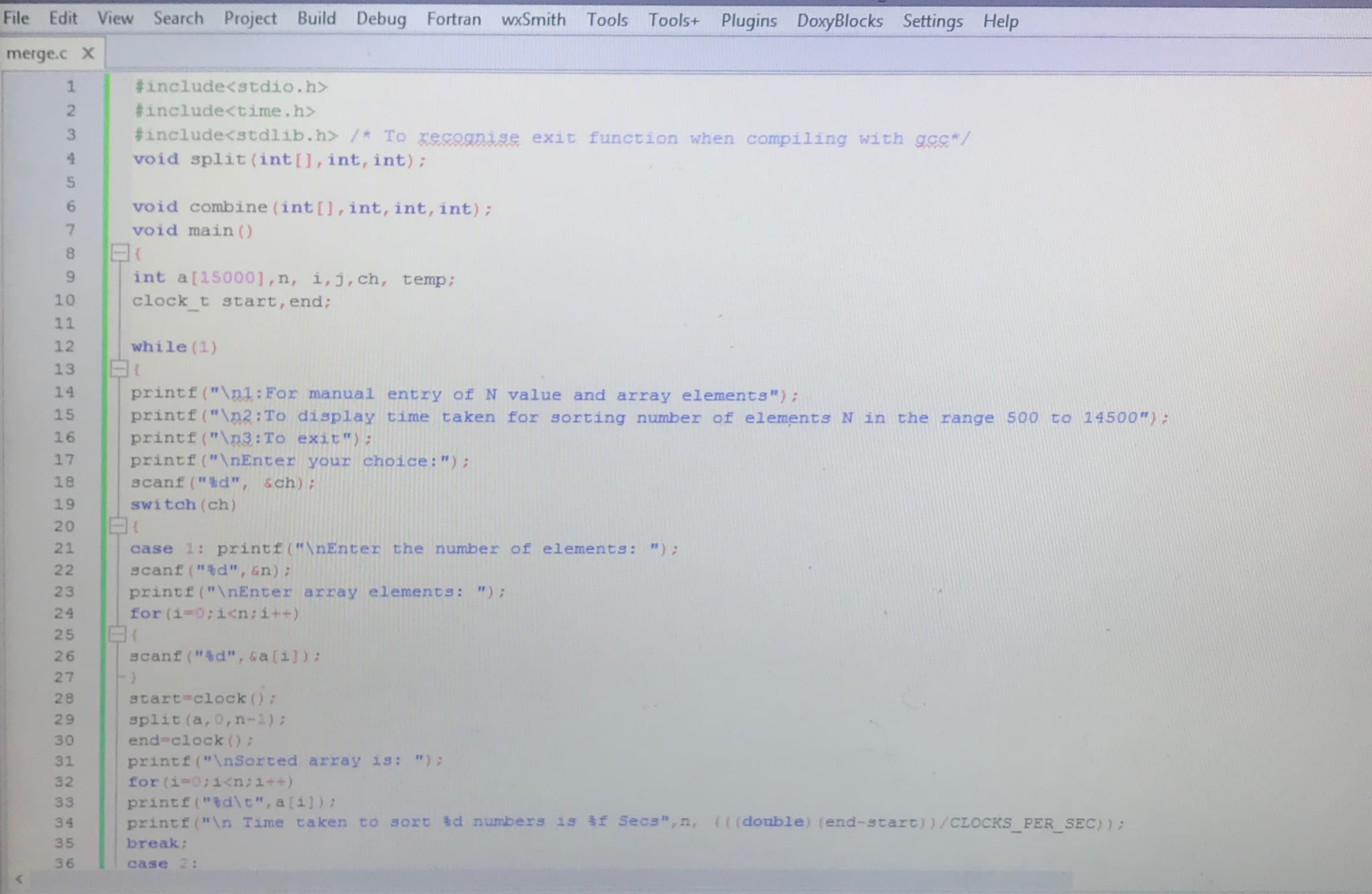
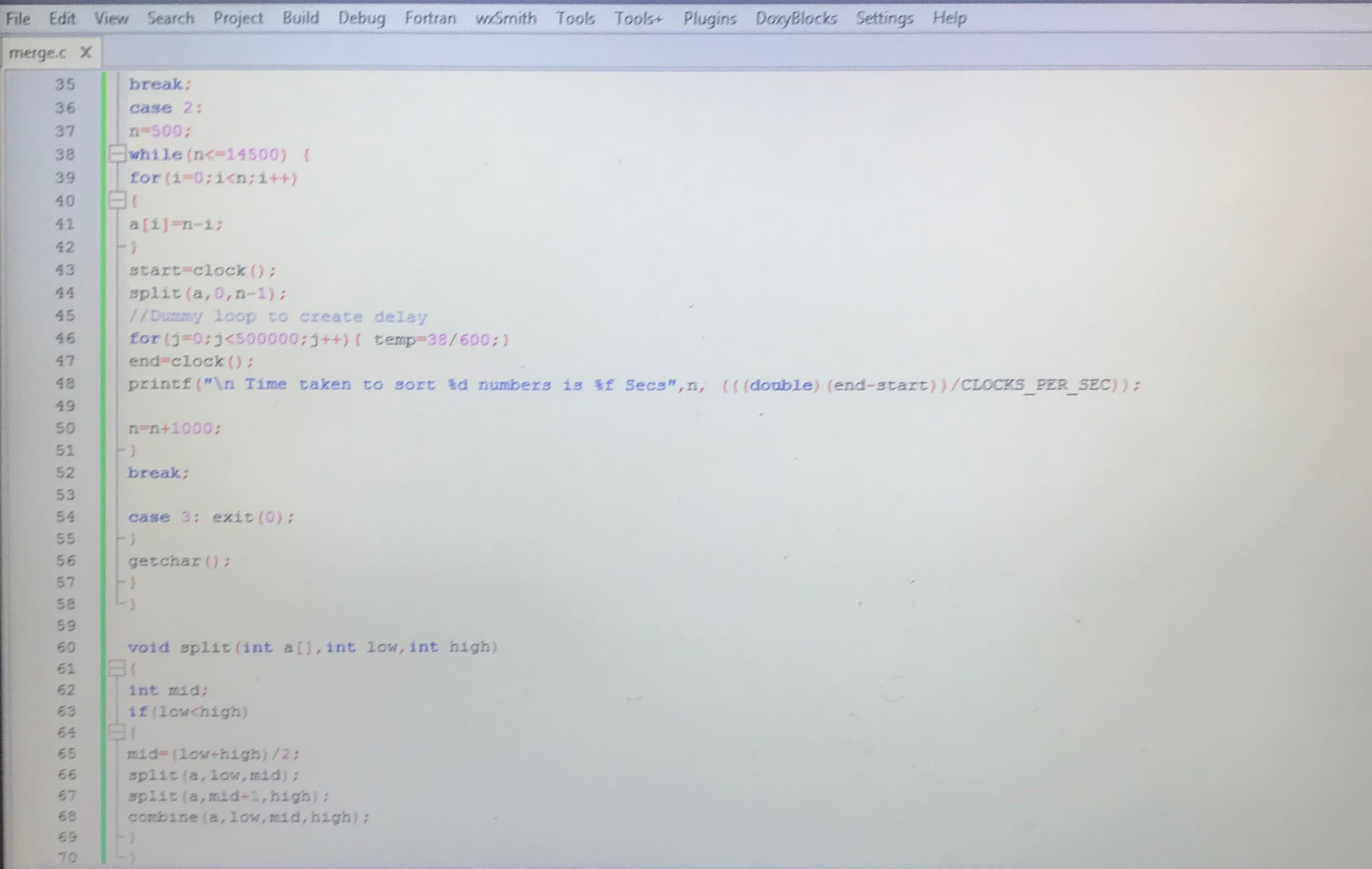
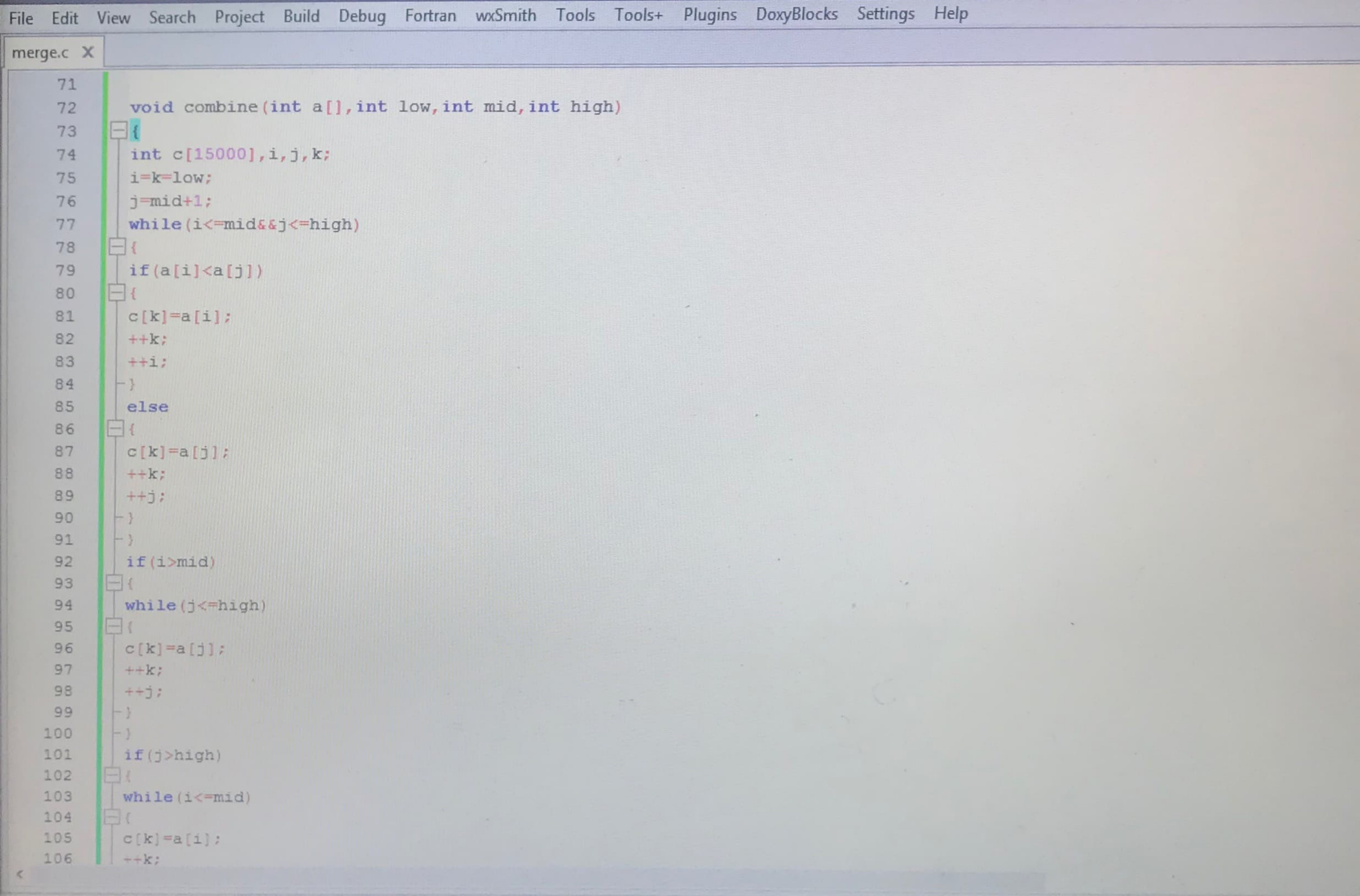


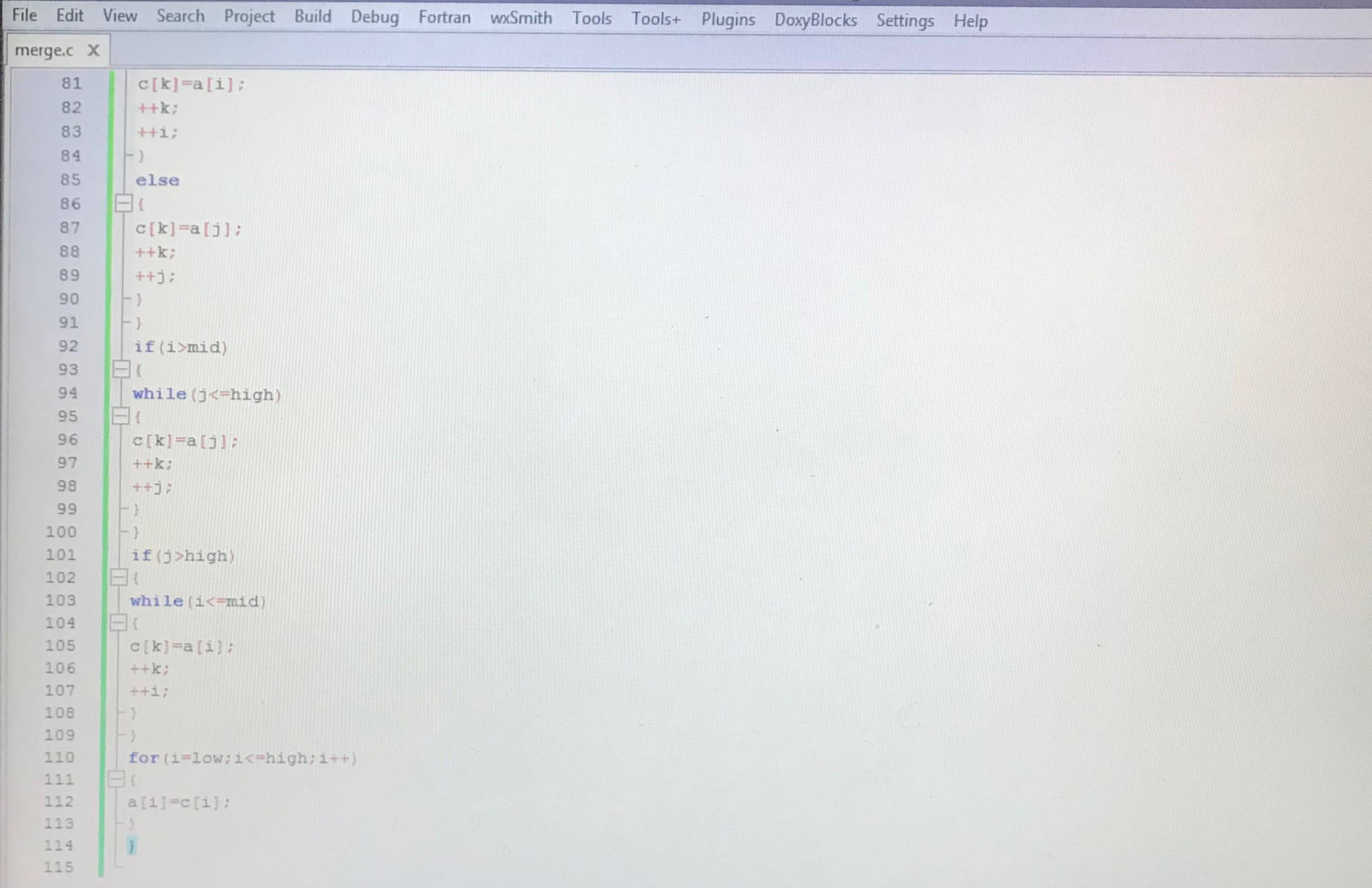
```
#include<stdio.h>
#include<time.h>
#include<stdlib.h> /* To recognise exit function when compiling with gcc*/
void split(int[],int,int);

void combine(int[],int,int,int);
void main()
{
int a[15000],n, i,j,ch, temp;
clock_t start,end;

while(1)
{
printf("\n1:For manual entry of N value and array elements");
printf("\n2:To display time taken for sorting number of elements N in the range 500 to 14500");
printf("\n3:To exit");
printf("\nEnter your choice:");
scanf("%d", &ch);
switch(ch)
{
case 1: printf("\nEnter the number of elements: ");
scanf("%d",&n);
printf("\nEnter array elements: ");
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
}
start=clock();
split(a,0,n-1);
end=clock();
printf("\nSorted array is: ");
for(i=0;i<n;i++)
printf("%d\t",a[i]);
printf("\n Time taken to sort %d numbers is %f Secs",n, (((double)(end-start))/CLOCKS_PER_SEC));
break;
case 2:
```



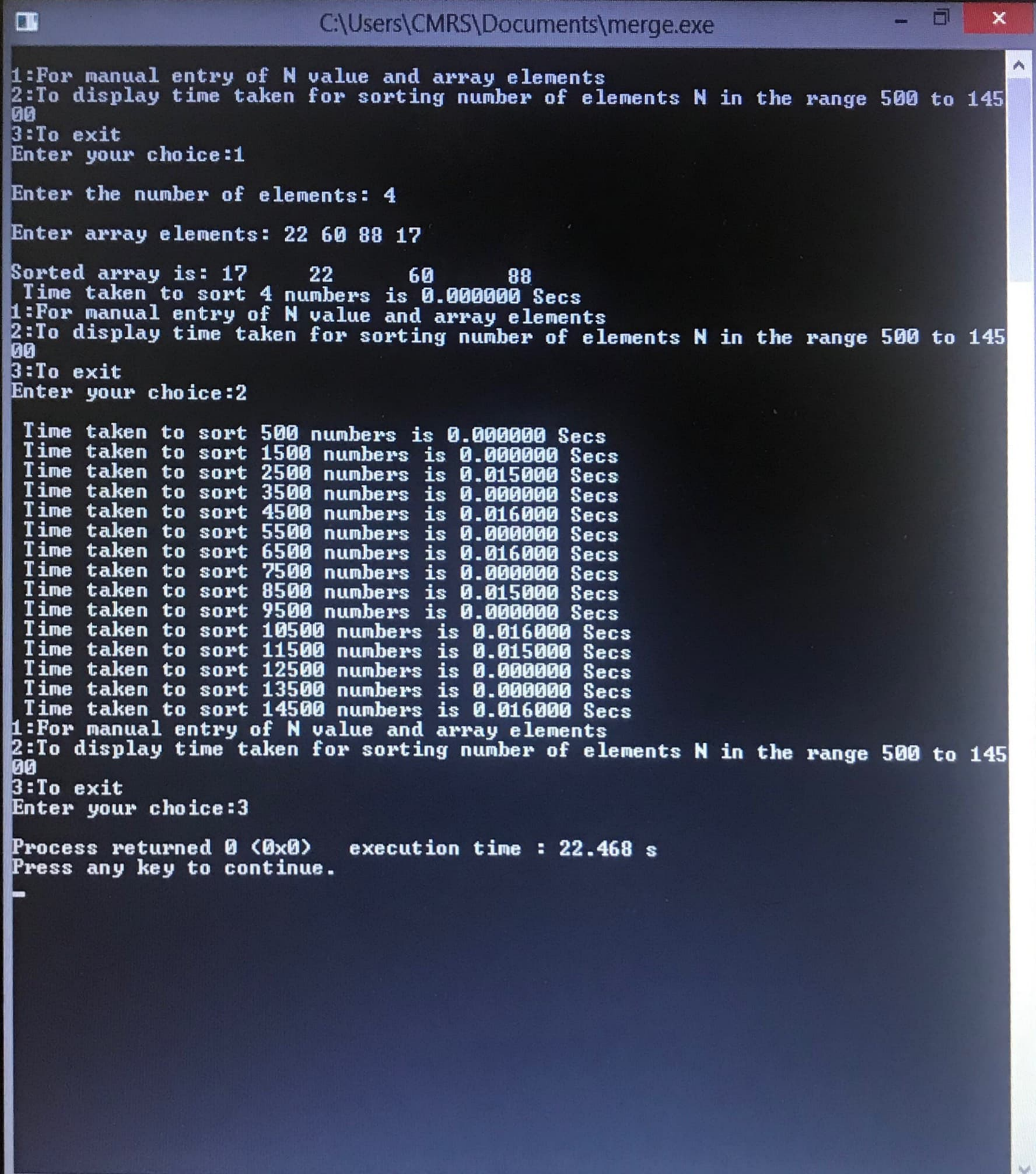
```c
break;
case 2:
n=500;
while(n<=14500) {
for(i=0;i<n;i++)
{
a[i]=n-i;
}
start=clock();
split(a,0,n-1);
//Dummy loop to create delay
for(j=0;j<500000;j++){ temp=38/600;}
end=clock();
printf("\n Time taken to sort %d numbers is %f Secs",n, (((double)(end-start))/CLOCKS_PER_SEC));

n=n+1000;
}
break;

case 3: exit(0);
}
getchar();
}
}

void split(int a[],int low,int high)
{
int mid;
if(low<high)
{
mid=(low+high)/2;
split(a,low,mid);
split(a,mid+1,high);
combine(a,low,mid,high);
}
}
```



```c

void combine(int a[],int low,int mid,int high)
{
int c[15000],i,j,k;
i=k=low;
j=mid+1;
while(i<=mid&&j<=high)
{
if(a[i]<a[j])
{
c[k]=a[i];
++k;
++i;
}
else
{
c[k]=a[j];
++k;
++j;
}
}
if(i>mid)
{
while(j<=high)
{
c[k]=a[j];
++k;
++j;
}
}
if(j>high)
{
while(i<=mid)
{
c[k]=a[i];
++k;
```



```
c[k]=a[i];
++k;
++i;
}
else
{
c[k]=a[j];
++k;
++j;
}
}
if(i>mid)
{
while(j<=high)
{
c[k]=a[j];
++k;
++j;
}
}
if(j>high)
{
while(i<=mid)
{
c[k]=a[i];
++k;
++i;
}
}
for(i=low;i<=high;i++)
{
a[i]=c[i];
}
}
```



```
1:For manual entry of N value and array elements
2:To display time taken for sorting number of elements N in the range 500 to 145
00
3:To exit
Enter your choice:1

Enter the number of elements: 4

Enter array elements: 22 60 88 17

Sorted array is: 17      22      60      88
 Time taken to sort 4 numbers is 0.000000 Secs
1:For manual entry of N value and array elements
2:To display time taken for sorting number of elements N in the range 500 to 145
00
3:To exit
Enter your choice:2

 Time taken to sort 500 numbers is 0.000000 Secs
 Time taken to sort 1500 numbers is 0.000000 Secs
 Time taken to sort 2500 numbers is 0.015000 Secs
 Time taken to sort 3500 numbers is 0.000000 Secs
 Time taken to sort 4500 numbers is 0.016000 Secs
 Time taken to sort 5500 numbers is 0.000000 Secs
 Time taken to sort 6500 numbers is 0.016000 Secs
 Time taken to sort 7500 numbers is 0.000000 Secs
 Time taken to sort 8500 numbers is 0.015000 Secs
 Time taken to sort 9500 numbers is 0.000000 Secs
 Time taken to sort 10500 numbers is 0.016000 Secs
 Time taken to sort 11500 numbers is 0.015000 Secs
 Time taken to sort 12500 numbers is 0.000000 Secs
 Time taken to sort 13500 numbers is 0.000000 Secs
 Time taken to sort 14500 numbers is 0.016000 Secs
1:For manual entry of N value and array elements
2:To display time taken for sorting number of elements N in the range 500 to 145
00
3:To exit
Enter your choice:3

Process returned 0 (0x0)   execution time : 22.468 s
Press any key to continue.
_
```