insertion_sort.c | johnson_trotter.c

insertion_sort.c

```c
/* Sort a given set of N integer elements using the Insertion sort technique and also compute timetake for different N values. */
#include<stdio.h>
#include <stdlib.h>
#include <time.h>
clock_t start, end;
double cpu_time;
int main()
{
    int i,j,temp,n,arr[10000],c,d;
    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();
    for(i=1;i<n;i++)
    {
        temp=arr[i];
        j=i-1;
        while(j>=0 && arr[j]>temp)
        {
            arr[j+1]=arr[j];
            j=j-1;
        }
        arr[j+1]=temp;
    }
    for (c = 1; c <= 5000; c++) for (d = 1; d <= 5000; d++) { }
    end = clock();
    cpu_time = (double)(end - start) / CLOCKS_PER_SEC;
    printf("\nSorted array is:\n");
    for(i=0;i<n;i++)
    {
        printf("%d ",arr[i]);
    }
     printf("\nExecution time for insertion sort = %f ms\n", cpu_time*1000);
```

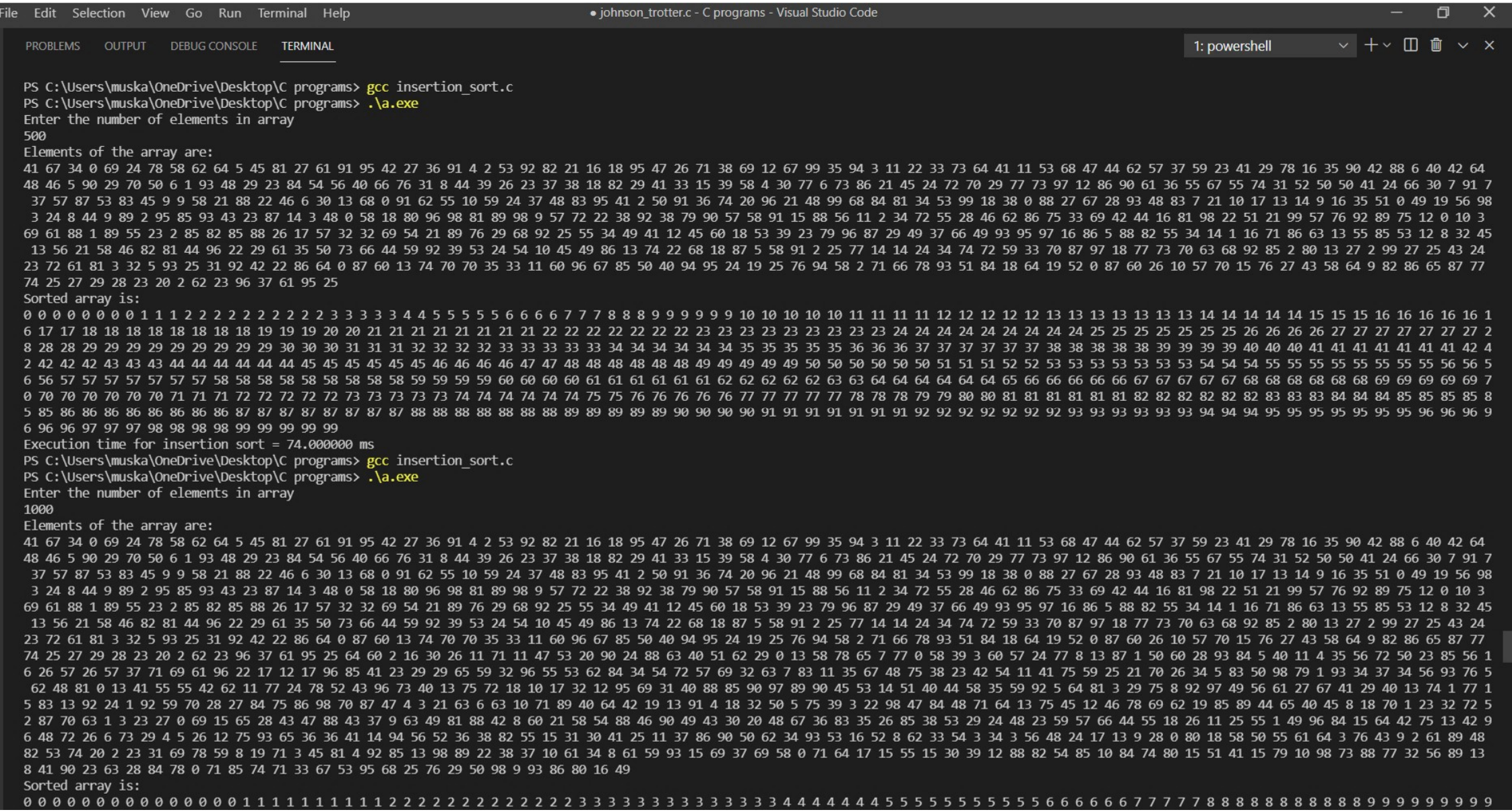
insertion_sort.c | johnson_trotter.c

insertion_sort.c

```c
        {
            temp=arr[i];
            j=i-1;
            while(j>=0 && arr[j]>temp)
            {
                arr[j+1]=arr[j];
                j=j-1;
            }
            arr[j+1]=temp;
        }
        for (c = 1; c <= 5000; c++) for (d = 1; d <= 5000; d++) { }
        end = clock();
        cpu_time = (double)(end - start) / CLOCKS_PER_SEC;
        printf("\nSorted array is:\n");
        for(i=0;i<n;i++)
        {
            printf("%d ",arr[i]);
        }
        printf("\nExecution time for insertion sort = %f ms\n", cpu_time*1000);
    }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: powershell

```
48 46 5 90 29 70 50 6 1 93 48 29 23 84 54 56 40 66 76 31 8 44 39 26 23 37 38 18 82 29 41 33 15 39 58 4 30 77 6 73 86 21 45 24 72 70 29 77 73 97 12 86 90 61 36 55 67 55 74 31 52 50 50 41 24 66 30 7 91 7
 37 57 87 53 83 45 9 9 58 21 88 22 46 6 30 13 68 0 91 62 55 10 59 24 37 48 83 95 41 2 50 91 36 74 20 96 21 48 99 68 84 81 34 53 99 18 38 0 88 27 67 28 93 48 83 7 21 10 17 13 14 9 16 35 51 0 49 19 56 98
 3 24 8 44 9 89 2 95 85 93 43 23 87 14 3 48 0 58 18 80 96 98 81 89 98 9 57 72 22 38 92 38 79 90 57 58 91 15 88 56 11 2 34 72 55 28 46 62 86 75 33 69 42 44 16 81 98 22 51 21 99 57 76 92 89 75 12 0 10 3
69 61 88 1 89 55 23 2 85 82 85 88 26 17 57 32 32 69 54 21 89 76 29 68 92 25 55 34 49 41 12 45 60 18 53 39 23 79 96 87 29 49 37 66 49 93 95 97 16 86 5 88 82 55 34 14 1 16 71 86 63 13 55 85 53 12 8 32 45
 13 56 21 58 46 82 81 44 96 22 29 61 35 50 73 66 44 59 92 39 53 24 54 10 45 49 86 13 74 22 68 18 87 5 58 91 2 25 77 14 14 24 34 74 72 59 33 70 87 97 18 77 73 70 63 68 92 85 2 80 13 27 2 99 27 25 43 24
23 72 61 81 3 32 5 93 25 31 92 42 22 86 64 0 87 60 13 74 70 70 35 33 11 60 96 67 85 50 40 94 95 24 19 25 76 94 58 2 71 66 78 93 51 84 18 64 19 52 0 87 60 26 10 57 70 15 76 27 43 58 64 9 82 86 65 87 77
74 25 27 29 28 23 20 2 62 23 96 37 61 95 25
Sorted array is:
0 0 0 0 0 0 0 0 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 4 4 5 5 5 5 5 6 6 6 6 7 7 7 8 8 8 9 9 9 9 9 9 10 10 10 10 10 11 11 11 11 12 12 12 12 12 13 13 13 13 13 13 13 14 14 14 14 14 15 15 15 16 16 16 16 16 1
6 17 17 18 18 18 18 18 18 18 18 19 19 19 20 20 21 21 21 21 21 21 21 21 21 22 22 22 22 22 22 22 23 23 23 23 23 23 23 23 23 24 24 24 24 24 24 24 24 25 25 25 25 25 25 25 26 26 26 26 27 27 27 27 27 27 27 2
8 28 28 29 29 29 29 29 29 29 29 29 30 30 30 31 31 31 32 32 32 32 33 33 33 33 33 34 34 34 34 34 34 35 35 35 35 35 36 36 36 37 37 37 37 37 37 38 38 38 38 38 39 39 39 39 40 40 40 41 41 41 41 41 41 41 42 4
```


File Edit Selection View Go Run Terminal Help
• johnson_trotter.c - C programs - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: powershell
PS C:\Users\muska\OneDrive\Desktop\C programs> gcc insertion_sort.c
PS C:\Users\muska\OneDrive\Desktop\C programs> .\a.exe
Enter the number of elements in array
500
Elements of the array are:
41 67 34 0 69 24 78 58 62 64 5 45 81 27 61 91 95 42 27 36 91 4 2 53 92 82 21 16 18 95 47 26 71 38 69 12 67 99 35 94 3 11 22 33 73 64 41 11 53 68 47 44 62 57 37 59 23 41 29 78 16 35 90 42 88 6 40 42 64
48 46 5 90 29 70 50 6 1 93 48 29 23 84 54 56 40 66 76 31 8 44 39 26 23 37 38 18 82 29 41 33 15 39 58 4 30 77 6 73 86 21 45 24 72 70 29 77 73 97 12 86 90 61 36 55 67 55 74 31 52 50 50 41 24 66 30 7 91 7
37 57 87 53 83 45 9 9 58 21 88 22 46 6 30 13 68 0 91 62 55 10 59 24 37 48 83 95 41 2 50 91 36 74 20 96 21 48 99 68 84 81 34 53 99 18 38 0 88 27 67 28 93 48 83 7 21 10 17 13 14 9 16 35 51 0 49 19 56 98
3 24 8 44 9 89 2 95 85 93 43 23 87 14 3 48 0 58 18 80 96 98 81 89 98 9 57 72 22 38 92 38 79 90 57 58 91 15 88 56 11 2 34 72 55 28 46 62 86 75 33 69 42 44 16 81 98 22 51 21 99 57 76 92 89 75 12 0 10 3
69 61 88 1 89 55 23 2 85 82 85 88 26 17 57 32 32 69 54 21 89 76 29 68 92 25 55 34 49 41 12 45 60 18 53 39 23 79 96 87 29 49 37 66 49 93 95 97 16 86 5 88 82 55 34 14 1 16 71 86 63 13 55 85 53 12 8 32 45
13 56 21 58 46 82 81 44 96 22 29 61 35 50 73 66 44 59 92 39 53 24 54 10 45 49 86 13 74 22 68 18 87 5 58 91 2 25 77 14 14 24 34 74 72 59 33 70 87 97 18 77 73 70 63 68 92 85 2 80 13 27 2 99 27 25 43 24
23 72 61 81 3 32 5 93 25 31 92 42 22 86 64 0 87 60 13 74 70 70 35 33 11 60 96 67 85 50 40 94 95 24 19 25 76 94 58 2 71 66 78 93 51 84 18 64 19 52 0 87 60 26 10 57 70 15 76 27 43 58 64 9 82 86 65 87 77
74 25 27 29 28 23 20 2 62 23 96 37 61 95 25
Sorted array is:
0 0 0 0 0 0 0 0 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 4 4 5 5 5 5 5 6 6 6 6 7 7 7 8 8 8 9 9 9 9 9 9 10 10 10 10 10 11 11 11 11 12 12 12 12 12 13 13 13 13 13 13 13 14 14 14 14 14 15 15 15 16 16 16 16 16 1
6 17 17 18 18 18 18 18 18 18 18 19 19 19 20 20 21 21 21 21 21 21 21 21 21 22 22 22 22 22 22 22 23 23 23 23 23 23 23 23 23 24 24 24 24 24 24 24 24 25 25 25 25 25 25 25 26 26 26 26 27 27 27 27 27 27 27 2
8 28 28 29 29 29 29 29 29 29 29 29 30 30 30 31 31 31 32 32 32 32 33 33 33 33 33 33 34 34 34 34 34 34 35 35 35 35 35 36 36 36 37 37 37 37 37 37 38 38 38 38 38 39 39 39 40 40 40 41 41 41 41 41 41 42 4
2 42 42 42 43 43 43 44 44 44 44 44 44 45 45 45 45 45 45 46 46 46 46 47 47 48 48 48 48 48 48 49 49 49 49 50 50 50 50 50 51 51 51 52 52 53 53 53 53 53 53 53 54 54 54 55 55 55 55 55 55 55 55 56 56 5
6 56 57 57 57 57 57 57 57 58 58 58 58 58 58 58 58 58 59 59 59 59 60 60 60 60 61 61 61 61 61 62 62 62 62 62 63 63 64 64 64 64 64 64 65 66 66 66 66 66 67 67 67 67 67 68 68 68 68 68 69 69 69 69 69 7
0 70 70 70 70 70 70 71 71 71 72 72 72 72 72 73 73 73 73 73 74 74 74 74 74 74 75 75 76 76 76 76 76 77 77 77 77 77 78 78 78 79 79 80 80 81 81 81 81 81 81 82 82 82 82 82 82 83 83 83 84 84 84 85 85 85 85 8
5 85 86 86 86 86 86 86 86 87 87 87 87 87 87 87 87 88 88 88 88 88 88 88 89 89 89 89 90 90 90 90 91 91 91 91 91 91 91 92 92 92 92 92 92 92 93 93 93 93 93 93 94 94 94 95 95 95 95 95 95 95 96 96 96 9
6 96 96 97 97 97 98 98 98 98 99 99 99 99 99
Execution time for insertion sort = 74.000000 ms
PS C:\Users\muska\OneDrive\Desktop\C programs> gcc insertion_sort.c
PS C:\Users\muska\OneDrive\Desktop\C programs> .\a.exe
Enter the number of elements in array
1000
Elements of the array are:
41 67 34 0 69 24 78 58 62 64 5 45 81 27 61 91 95 42 27 36 91 4 2 53 92 82 21 16 18 95 47 26 71 38 69 12 67 99 35 94 3 11 22 33 73 64 41 11 53 68 47 44 62 57 37 59 23 41 29 78 16 35 90 42 88 6 40 42 64
48 46 5 90 29 70 50 6 1 93 48 29 23 84 54 56 40 66 76 31 8 44 39 26 23 37 38 18 82 29 41 33 15 39 58 4 30 77 6 73 86 21 45 24 72 70 29 77 73 97 12 86 90 61 36 55 67 55 74 31 52 50 50 41 24 66 30 7 91 7
37 57 87 53 83 45 9 9 58 21 88 22 46 6 30 13 68 0 91 62 55 10 59 24 37 48 83 95 41 2 50 91 36 74 20 96 21 48 99 68 84 81 34 53 99 18 38 0 88 27 67 28 93 48 83 7 21 10 17 13 14 9 16 35 51 0 49 19 56 98
3 24 8 44 9 89 2 95 85 93 43 23 87 14 3 48 0 58 18 80 96 98 81 89 98 9 57 72 22 38 92 38 79 90 57 58 91 15 88 56 11 2 34 72 55 28 46 62 86 75 33 69 42 44 16 81 98 22 51 21 99 57 76 92 89 75 12 0 10 3
69 61 88 1 89 55 23 2 85 82 85 88 26 17 57 32 32 69 54 21 89 76 29 68 92 25 55 34 49 41 12 45 60 18 53 39 23 79 96 87 29 49 37 66 49 93 95 97 16 86 5 88 82 55 34 14 1 16 71 86 63 13 55 85 53 12 8 32 45
13 56 21 58 46 82 81 44 96 22 29 61 35 50 73 66 44 59 92 39 53 24 54 10 45 49 86 13 74 22 68 18 87 5 58 91 2 25 77 14 14 24 34 74 72 59 33 70 87 97 18 77 73 70 63 68 92 85 2 80 13 27 2 99 27 25 43 24
23 72 61 81 3 32 5 93 25 31 92 42 22 86 64 0 87 60 13 74 70 70 35 33 11 60 96 67 85 50 40 94 95 24 19 25 76 94 58 2 71 66 78 93 51 84 18 64 19 52 0 87 60 26 10 57 70 15 76 27 43 58 64 9 82 86 65 87 77
74 25 27 29 28 23 20 2 62 23 96 37 61 95 25 64 60 2 16 30 26 11 71 11 47 53 20 90 24 88 63 40 51 62 29 0 13 58 78 65 7 77 0 58 39 3 60 57 24 77 8 13 87 1 50 60 28 93 84 5 40 11 4 35 56 72 50 23 85 56 1
6 26 57 26 57 37 71 69 61 96 22 17 12 17 96 85 41 23 29 29 65 59 32 96 55 53 62 84 34 54 72 57 69 32 63 7 83 11 35 67 48 75 38 23 42 54 11 41 75 59 25 21 70 26 34 5 83 50 98 79 1 93 34 37 34 56 93 76 5
62 48 81 0 13 41 55 55 42 62 11 77 24 78 52 43 96 73 40 13 75 72 18 10 17 32 12 95 69 31 40 88 85 90 97 89 90 45 53 14 51 40 44 58 35 59 92 5 64 81 3 29 75 8 92 97 49 56 61 27 67 41 29 40 13 74 1 77 1
5 83 13 92 24 1 92 59 70 28 27 84 75 86 98 70 87 47 4 3 21 63 6 63 10 71 89 40 64 42 19 13 91 4 18 32 50 5 75 39 3 22 98 47 84 48 71 64 13 75 45 12 46 78 69 62 19 85 89 44 65 40 45 8 18 70 1 23 32 72 5
2 87 70 63 1 3 23 27 0 69 15 65 28 43 47 88 43 37 9 63 49 81 88 42 8 60 21 58 54 88 46 90 49 43 30 20 48 67 36 83 35 26 85 38 53 29 24 48 23 59 57 66 44 55 18 26 11 25 55 1 49 96 84 15 64 42 75 13 42 9
6 48 72 26 6 73 29 4 5 26 12 75 93 65 36 36 41 14 94 56 52 36 38 82 55 15 31 30 41 25 11 37 86 90 50 62 34 93 53 16 52 8 62 33 54 3 34 3 56 48 24 17 13 9 28 0 80 18 58 50 55 61 64 3 76 43 9 2 61 89 48
82 53 74 20 2 23 31 69 78 59 8 19 71 3 45 81 4 92 85 13 98 89 22 38 37 10 61 34 8 61 59 93 15 69 37 69 58 0 71 64 17 15 55 15 30 39 12 88 82 54 85 10 84 74 80 15 51 41 15 79 10 98 73 88 77 32 56 89 13
8 41 90 23 63 28 84 78 0 71 85 74 71 33 67 53 95 68 25 76 29 50 98 9 93 86 80 16 49
Sorted array is:
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 7 7 7 7 8 8 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9

N	TIME(ms)
500	70
1000	77
1500	75
2000	71
2500	82
3000	80