

```
#include <stdio.h>
int main() {
```

```
#include <stdio.h>
int main() {
```

main()

```

int a[5] = {5, 4, 1, 3, 2}; m, n, temp;
printf("Before Sorting - |n");
for (m=0; m<5; m++)
    printf("%d\t", a[m]);

for (m=0; m<4; m++)
    for (n=0; n<4-m; n++)
        if (a[n] > a[n+1])
            temp = a[n];
            a[n] = a[n+1];
            a[n+1] = temp;

printf("After Sorting - |n");
for (m=0; m<5; m++)
    printf("%d\t", a[m]);

return 0;

```

After Sorting

<u>m</u>	<u><math>m \leq 5</math></u>	<u><math>a[m]</math></u>
0	$0 \leq 5$ (T)	1
1	$1 \leq 5$ (T)	2
2	$2 \leq 5$ (T)	3
3	$3 \leq 5$ (T)	4
4	$4 \leq 5$ (T)	5
5	$5 \leq 5$ (F) $\rightarrow$ X	

Before Sorting...

	0	1	2	3	4
0	5	4	1	3	2

$m$        $m < 5$

0      0 < 5 (T)

1      1 < 5 (T)

2      2 < 5 (T)

3      3 < 5 (T)

4      4 < 5 (T)

5      5 < 5 (F)

$\binom{m=0}{0 \leq 4}$	$\binom{n=0}{0}$	$\frac{n \leq 4}{0 \leq 4 - (0 \leq 4)}$
	/	
	1	$1 \leq 4 - (1 \leq 4)$
	2	$2 \leq 4 - (2 \leq 4)$
	3	$3 \leq 4 - (3 \leq 4)$

$$\begin{pmatrix} m=1 \\ (2 \times 4) \\ T \end{pmatrix} \quad 0$$
$$\begin{array}{ll} 1 & 1 < 4- \\ & (1 < 2) \\ 2 & 2 < 4- \\ & (2 < 3) \\ 3 & 3 < 4-1 \\ & (3 < 3) \end{array}$$
$$\begin{array}{ccc} m=2 & & 0 < 4- \\ 2 < 4 & & (0 < 2) \\ (T) & & \end{array}$$
$$0 \leq y \leq 4$$
$$\begin{pmatrix} 4 \\ 4 \\ 1 \end{pmatrix} = -X$$

$m =$

0	1	2	3	
---	---	---	---	--

$n =$

0	0	0	0
1	1	1	1
2	2	2	
3	3		
4			

```
temp = a[n];  
a[n] = a[n+1];  
a[n+1] = temp;  
temp = a[0]; // temp = 5;  
a[0] = a[0+1]; // 5 = 4;  
a[1] = temp; // 4 = temp;  
temp = a[1]; // temp = 5;  
a[1] = a[1+1]; // 5 = 1;  
a[2] = temp; // 1 = temp;  
temp = a[2]; // temp = 5;  
a[2] = a[2+1]; // 5 = 3;  
a[3] = temp; // 3 = temp;  
temp = a[3]; // temp = 5;  
a[3] = a[3+1]; // 5 = 2;  
a[4] = temp; // 2 = temp;
```

```
temp = a[0]; // temp = 4;  
a[0] = a[0+1]; // 4 = 1;  
a[1] = temp; // 1 = temp  
  
temp = a[1]; // temp = 4;  
a[1] = a[1+1]; // 4 = 3;  
a[2] = temp; // 3 = temp;  
  
temp = a[2]; // temp = 4;  
a[2] = a[2+1]; // 4 = 2;  
a[3] = temp; // 2 = temp
```

```
temp = a[1]; // temp = 3;  
a[1] = a[1+1]; // 3 = 2;  
a[2] = temp; // 2 = temp;
```

→ output screen :-

Before Sorting - -  
5 4 1 3 2

After Sorting - -  
1 2 3 4 5

Updating Array after sorting :-

0	1	2	3
4	5	1	3

0	1	2	3	4
4	1	5	3	2

  

0	1	2	3	4
			5	2

4 1 3

0 1 2 3 4

4 1 3 2 5

unsorted

0	1	2	3	4
				5

1	1	3	4	
---	---	---	---	--

  

0	1	2	3	4
1	3	4	2	5

0 1 2 3 4

1	3	2	4	5
---	---	---	---	---

unsorted sorted

0 1 2 3 4  
1 3 2 4 5

0 1 2 3 4  
1 2 3 4 5

1	2	3	4	5
---	---	---	---	---

pratiksha panda.  
Sec-B  
Regno- 2001297156