

```
1 #include<stdio.h>
2 #define s 10
3 #define N 50
4
5 int data[N], next[N], head[s], i=0;
6
7 int h(int k){return k%s; }
8
9 void in(int k)
10 {
11     data[i] = k;
12     next[i] = head[h(k)];
13     head[h(k)] = i;
14     i++;
15 }
16
17 void dis()
18 {
19     for(int j=0;j<5;j++)
20     {
21         printf("%d: ",j);
22         for( int x = head[j]; x!= -1; x= next[x])
23             printf("%d->",data[x]);
24         printf("NULL\n");
25     }
26 }
27
28 int main()
29 {
30     for(int j=0;j<s;j++) head[j] = -1;
31     in(10); in(20); in(15); in(25); in(35);
32     dis();
33 }
34
```

Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\sreel\OneDrive\Desktop\Untitled1.exe
- Output Size: 129.2236328125 KiB
- Compilation Time: 1.28s

```
0: 20->10->NULL
1: NULL
2: NULL
3: NULL
4: NULL
```

Process exited after 0.0588 seconds with return value 0
Press any key to continue . . . |

```
1 #include<stdio.h>
2 #define SIZE 5
3 #define MAX 20
4 int data[MAX], next[MAX], head[SIZE], i=0;
5 int hash(int key) {return key %SIZE;}
6 void insert(int key)
7 {
8     data[i] = key;
9     next[i] = head[hash(key)];
10    head[hash(key)] = i;
11    i++;
12 }
13 void display()
14 {
15     for(int j=0;j<SIZE;j++)
16     {
17         printf("index %d: ",j);
18         for(int x = head[j]; x!= -1; x = next[x])
19             printf("%d->", data[x]);
20         printf("NULL\n");
21     }
22 }
23 int main()
24 {
25     for(int j = 0; j<SIZE; j++) head[j] = -1;
26
27     insert(7);
28     insert(12);
29     insert(17);
30     insert(22);
31     insert(32);
32
33     display();
34 }
```

```
index 0: NULL
index 1: NULL
index 2: 32->22->17->12->7->NULL
index 3: NULL
index 4: NULL
```

```
Process exited after 0.06805 seconds with return value 0
Press any key to continue . . . |
```

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int a[5] = {64, 25, 12, 22, 11}, n=5,i,j,min,temp;
6     for(i=0;i<n;i++)
7     {
8         min=i;
9         for(j=i+1;j<n;j++)
10            if(a[j]<a[min]) min =j;
11
12         temp = a[i];
13         a[i] = a[min];
14         a[min] = temp;
15     }
16
17     for(i=0;i<n;i++)
18     printf("%d",a[i]);
19
20     return 0;
21 }
```

```
1112222564
```

```
-----  
Process exited after 0.06994 seconds with return value 0  
Press any key to continue . . . |
```

```
1 #include<stdio.h>
2 void heapify(int a[], int n, int i)
3 {
4     int largest = i, l = 2*i+1, r = 2*i+2, temp;
5     if(l < n && a[r] > a[largest]) largest = l;
6     if(r < n && a[r] > a[largest]) largest = r;
7
8     if (largest != i)
9     {
10         temp = a[i];
11         a[i] = a[largest];
12         a[largest] = temp;
13         heapify(a, n, largest);
14     }
15 }
16 void heapsort(int a[], int n){
17     for(int i=n/2-1;i>=0;i--)
18         heapify(a, n, i);
19     for(int i = n-1;i>=0;i--)
20     {
21         int temp = a[0];
22         a[0] = a[i];
23         a[i] = temp;
24         heapify(a, i, 0);
25     }
26 }
27 int main(){
28     int a[6] = {40, 10, 30, 50, 20, 60}, n=6;
29     heapsort(a, n);
30     for(int i=0;i<n;i++)
31         printf("%d", a[i]);
32     return 0;
33 }
```

201030506040

Process exited after 0.08166 seconds with return value 0
Press any key to continue . . . |

```
1 #include<stdio.h>
2
3 void quick(int a[], int low, int high)
4 {
5     if(low < high)
6     {
7         int i = low, j=high, pivot = a[low],temp;
8         while(i < j)
9         {
10             while(a[i] <= pivot && i < high)
11                 i++;
12             while(a[j] > pivot) j--;
13             if(i < j)
14             {
15                 temp = a[i];
16                 a[i] = a[j];
17                 a[j] = temp;
18             }
19         }
20         a[low] = a[j];
21         a[j] = pivot;
22         quick(a,low,j-1);
23         quick(a,j+1,high);
24     }
25 }
26 int main()
27 {
28     int a[6] = {33, 10, 55, 71, 29, 3},n=6,i;
29     quick(a,0,n-1);
30     for(i=0;i<n;i++)
31     printf("%d",a[i]);
32     return 0;
33 }
```

31029335571

Process exited after 0.07106 seconds with return value 0
Press any key to continue . . . |

```
1 #include<stdio.h>
2
3 int getmax(int a[], int n)
4 {
5     int max = a[0];
6     for(int i=1;i<n;i++)
7         if(a[i]>max) max=a[i];
8     return max;
9 }
10
11 void countsort(int a[], int n, int exp)
12 {
13     int output[100], count[10] = {0};
14     for(int i=0;i<n;i++)
15         count[(a[i]/exp)%10]++;
16     for(int i=1;i<10;i--)
17         output[--count[(a[i]/exp)%10]] = a[i];
18     for(int i=0;i<n;i++)
19         a[i] = output[i];
20 }
21
22 int main()
23 {
24     int a[6] = {170, 45, 75, 90, 802, 24}, n=6;
25     int max = getmax(a,n);
26     for(int exp=1; max/exp>0;exp*=10)
27         countsort(a,n,exp);
28     for(int i=0;i<n;i++)
29         printf("%d",a[i]);
30     return 0;
31 }
```

```
Process exited after 2.843 seconds with return value 3221225477
Press any key to continue . . . |
```

```
1 #include<stdio.h>
2
3 void merge(int a[], int l, int m, int r)
4 {
5     int i=l, j=m+1, k=0, b[100];
6     while(i<=m && j<=r)
7         b[k++] = (a[i]<a[j]) ? a[i++] : a[j++];
8     while(i<=m) b[k++] = a[i++];
9     while(j<=r) b[k++] = a[j++];
10    for(i=0;i<k;i++) a[l+i] = b[i];
11 }
12
13 void mergesort(int a[], int l, int r)
14 {
15     if(l<r)
16     {
17         int m = (l+r)/2;
18         mergesort(a,l,m);
19         mergesort(a,m+1,r);
20         merge(a,l,m,r);
21     }
22 }
23
24 int main()
25 {
26     int a[6] = {38, 27, 43, 3, 9, 82}, n=6,i;
27     mergesort(a,0,n-1);
28     for(i=0;i<n;i++)
29         printf("%d",a[i]);
30     return 0;
31 }
```

4343434343971971120

Process exited after 0.07239 seconds with return value 0
Press any key to continue . . . |

```
1 #include<stdio.h>
2 #define SIZE 10
3
4 int hashtable[SIZE];
5
6 void insert(int key)
7 {
8     int index = key % SIZE;
9     while(hashtable[index] != -1)
10        index = (index + 1) % SIZE;
11     hashtable[index] = key;
12 }
13
14 void display()
15 {
16     for(int i=0;i<SIZE;i++)
17         printf("%d",hashtable[i]);
18 }
19
20 int main()
21 {
22     for(int i=0;i<SIZE;i++)
23         hashtable[i] = -1;
24
25     int data[] = {23, 43, 13, 27};
26     for(int i=0;i<4; i++)
27         insert(data[i]);
28
29     display();
30     return 0;
31 }
```

-1-1-1234313-127-1-1

Process exited after 0.06941 seconds with return value 0
Press any key to continue . . . |

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int a[] = {10, 20, 30, 40, 50};
6     int n=5, pos = 3;
7
8
9     for(int i=pos-1;i<n-1;i++)
10    {
11        a[i] = a[i+1];
12
13    n--;
14
15    for(int i=0;i<n;i++)
16        printf("%d",a[i]);
17
18    return 0;
19 }
```

10204050

Process exited after 0.07423 seconds with return value 0
Press any key to continue . . . |

```
1 #include<stdio.h>
2
3     int main()
4 {
5         int a[10] = {1, 2, 3, 5}, n=4, pos = 3, value = 4;
6
7
8         for(int i=0;i>=pos;i--)
9         {
10            a[i] = a[i-1];
11
12            a[pos-1] = value;
13            n++;
14        }
15        for(int i=0;i<n;i++)
16            printf("%d", a[i]);
17
18    return 0;
19 }
```

1235

Process exited after 0.07337 seconds with return value 0
Press any key to continue . . . |

```
[ ] Untitled1.cpp [ ] Untitled2.cpp Untitled3.cpp
1 #include<stdio.h>
2
3     int main()
4 {
5         int a[] = {10,20,30,40,50};
6         int n = 5, key = 30, i;
7
8         for(i=0;i<n;i++)
9         {
10             if(a[i] == key)
11             {
12                 printf("found at index %d\n",i);
13                 return 0;
14             }
15         }
16
17         printf("not found\n");
18         return 0;
19 }
```

```
found at index 2
```

```
-----
Process exited after 0.0642 seconds with return value 0
Press any key to continue . . .
```

```
1 #include<stdio.h>
2
3     int main()
4 {
5     int a[] = {5, 10, 15, 20}, n=4,i;
6     |
7
8     for(i=0;i<n;i++)
9     {
10         printf("%d",a[i]);
11     }
12     return 0;
13 }
```

```
5101520
```

```
-----  
Process exited after 0.0619 seconds with return value 0  
Press any key to continue . . .
```