Sort a given N integers using Quick sort technique and compute its time taken.

```
#include <stdio.h>
#include <stdlib.h>
void merge(int low,int mid,int high,int array[20],int mer[20])
{
int i = low;
int j = mid+1;
int k = 0;
while(i<=mid && j<=high)
{
if(array[i]<array[j])</pre>
mer[k] = array[i];
i++;
k++;
}
else
{
mer[k] = array[j];
j++;
k++;
}
}
while (i <= mid)
mer[k] = array[i];
i++;
k++;
}
```

```
while (j <= high)
mer[k] = array[j];
j++;
k++;
for(int i=0;i<k;i++)
{
array[low+i] = mer[i];
}
}
void merge_sort(int low,int high,int array[20],int merged[20])
{
if(low<high)
int mid = (low+high)/2;
merge_sort(low,mid,array,merged);
merge_sort(mid+1,high,array,merged);
merge(low,mid,high,array,merged);
}
int main()
int n,array[30];
printf("Enter no of elements:");
scanf("%d",&n);
printf("Enter elements:");
for(int i=0;i<n;i++)</pre>
```

```
scanf("%d",&array[i]);
int merged[30];
merge_sort(0,n-1,array,merged);
printf("Sorted array:");
for(int i=0;i<n;i++)
{
printf("%d ",array[i]);
}
}
#include <stdio.h>
void swap(int *a, int *b) {
int t = *a;
*a = *b;
*b = t;
int partition(int a[], int I, int h) {
int pivot = a[l];
int i = l, j = h;
while (i < j) {
while (a[i] <= pivot && i <= h) {
i++;
}
while (a[j] > pivot) {
j--;
}
if (i < j) {
swap(&a[i], &a[j]);
}
```

```
}
swap(&a[I], &a[j]);
return j;
}
void quickSort(int a[], int I, int h) {
if (I < h) {
int pi = partition(a, l, h);
quickSort(a, I, pi - 1);
quickSort(a, pi + 1, h);
}
}
int main() {
int a[20], n, i;
printf("Enter size of array\n");
scanf("%d", &n);
printf("Enter data elements: ");
for (i = 0; i < n; i++) {
scanf("%d", &a[i]);
}
printf("Unsorted Array\n");
for (i = 0; i < n; i++) {
printf("%d\t", a[i]);
}
quickSort(a, 0, n - 1);
printf("\nSorted array in ascending order: \n");
for (i = 0; i < n; i++) {
printf("%d\t", a[i]);
}
return 0;
```

```
Enter size of array

5
Enter data elements: 88 -5 65 -10 0 25 18
Unsorted Array

88 -5 65 -10 0

Sorted array in ascending order:
-10 -5 0 65 88

Process returned 0 (0x0) execution time: 22.359 s

Press any key to continue.
```

Implement 0/1 Knapsack problem using dynamic programming

```
#include <stdio.h>
#include <stdbool.h>
int p[15],w[15],maxW;
void main(){
  int n,i,j,maxP;
  printf("Enter the number of items\n");
  scanf("%d",&n);
  printf("Enter the max weight\n");
  scanf("%d",&maxW);
  printf("Enter the weights\n");
  for(i=0;i<n;i++)
  scanf("%d",&w[i]);
  printf("Enter the profits\n");
  for(i=0;i<n;i++)
  scanf("%d",&p[i]);
  maxP=knapsack(n);
  printf("Optimal profit is %d ",maxP);
```

```
}
```

```
int knapsack(int n) {
  int v[n+1][maxW+1],i,j;
  for (int i = 0; i \le n; i++) {
    for (int j = 0; j \le maxW; j++) {
       if (i == 0 | | j == 0)
         v[i][j] = 0;
       else if (w[i-1] \le j)
         v[i][j] = max(p[i-1] + v[i-1][j-w[i-1]], v[i-1][j]);
       else
         v[i][j] = v[i - 1][j];
    }
  }
  int selected[n];
  i = n; j = maxW;
  int count = 0;
  while (i > 0 \&\& j > 0) {
    if (v[i][j] != v[i - 1][j]) {
       selected[count++] = i;
       j -= w[i - 1];
       i--;
     } else {
       i--;
```

```
}
  }
  printf("TABLE \n");
  for (int i = 0; i \le n; i++) {
    for (int j = 0; j \le maxW; j++) {
       printf("%d ",v[i][j]);
    }
    printf("\n");
  }
  printf("Selected objects: ");
  for (int j = count - 1; j >= 0; j--)
    printf("%d ", selected[j]);
  printf("\n");
  return v[n][maxW];
}
int max(int a, int b) {
  return (a > b) ? a : b;
}
```

C:\Users\lenovo\Desktop\ADA\knapsack.exe

```
Enter the number of items
3
Enter the max weight
5
Enter the weights
2
4
6
Enter the profits
10
20
30
TABLE
0 0 0 0 0
0 10 10 10 10
0 0 10 10 20
0 0 10 10 20
Selected objects: 2
Optimal profit is 20
Process returned 21 (0x15) execution time : 15.415 s
Press any key to continue.
```

Sort a given set of N integers using puick sort technic
and compute its time taken.
include Lstdio. h>
include < Kime · h>
void quicksort lint number [25], int first, int late
inti, j pivot, temp; (as in. vii)
if (first zlast) (++) (azi dilla)
2
pivot = first; :(":100.19191210)"1 -/10111
1= Hirst; 2000 1100 1100 1000 1000 10002
-
- while (icj) ; coed bangar tol
\$ ((baprent, 1101110, 100, 21,1102, 2000)
while Inumber [1] = number [pivot] & & iz lost]
1++.
while (number [j] >number [pivot])
j;
- (12)
\$
temp = number [i];
number (1°) = pumbed().
3 Center Claration (1910)
FULET ETENEDIC
temp=number [pivot]; 21 100mg/s 1310
number Epivor) = number (3); 01 0 01
number Cf) = temp;
quicksort (number, first, J-1);
quicksort (number, jt1, 1act);
12 (ast);
2
lint main()
int maine?
Scanned with CamScanner

	Page ZI
int 1 count purel - 10 = 7	
int 1, count, number[25];	- Inama god
Start, end:	12 2 Shillian to
Start = clock();	12 shuhai at 1
Prent Wenter no of elements	Digrad him
Scant (117.011, & count);	· to 1 com tal
Prientfl" Enter y d elements: ", c	ount);
for 18 =0; 12 count; 1++)	L Course piev
Scanfi"/d", finumber[1]);	· ·
quick sort inumber, 0, count -1);	himit the
end = clock();	Starti, ya.
Print 1" Order of sorted element	Range A Augal
tor (1=0: 15 count : 1++) ++ 0==	101 11 = 1 1 12
Print fi"./.d", number (13);	in the second
Print + ("In time taten to sort pr.	1. f" ditations
(CD) STUTE) (COCKS-per-sec);	The same of the
Terurno; must doos to ritory adia	210) (1) frain)
3 (++1,0=)	1:1=11.0
	4
Output (Ci) 74	101 21 4 mois
ther the number of elements to be	2 corted: 5
Enter 5 elements som and rains	mal" Hapiff
10 (m S 1)	Deany (14)
89	* Knapsa (KT):
Take the second	and (! dalah
50	and the manufacture of the same of the sam
7-11/21/21	Void Kingsons
Order of corded elements: 171	0 58 8 89
time taken to sort is: 10:594000	1 017 11
Process returned 0 (0x0) execution	ntimo: 10.594 serc.
	24 (Q. 1)
(TT) 301	= 51 (3-711)

(.)	Dails	V
(Pode 55	0
0		1

n Gi	Page 22	20
	Implement O/1 Knapsack problem using dynamic from	35
	#include < stdio. h>	4
	# include Lonio.h>	
	Wid Kmosack();	
	lest some l'est let):	
	int i, j, n, m, p[10], w[10], v[10] [10];	
	void main()	
	\$ (Cilindrala D. 11 +anz	
	Printf ("Enter no. of items \032); mal +102 your	
	Scant (" Y.d", Gn);	
	Printy (" Enter the weight of each item: \n");	
	for li=1; i = n; i++) + +++++++++++++++++++++++++++++++	
	(11) reduing (10.1.1) + tall a	
	Scant (" Y.d", AW [i]) galat grait of " 1 + cars	
_	2000 - 100 -	
<u> </u>	Print (" Enter the profit of each item 10");	
	for (1°=1; 1' = n; 1++)	_
	Scanf [" 1.0", &p[i]);	
Si.	3 + 100 + 10 + 100	_
11	THE PARTY OF THE P	_
1	Printf ("In enter the knapsack's capacity:\t"); Scant ("/d", & m);	_
	Knapsa (KI);	_
	Jetch ();	_
	2	_
1	void Knapsack ()	_
	5 -3 -1 11 F 1 - 147900 415 120 h 2 -2 -2	_
	ntm[10];	_
	Proce 1 sturned 01010) , (title 21 20 21) 10	_
	or (j=0; jc=m; j++)	-
	The second secon	1
	(i==0115==0)	1
-CS Scan	ned with CamScapper	1

-	-bh	Classmate Date Page 23
	v [i][j]=0;	
	3	A STORY OF A STORY
	elese it 1j-wrizes	1-21 1=110
	II (V)	a secondo
		1/ DA" 104.1
	3 C1JCj-1Jv=C1-1Jcj3;	1
		Stages House Land
	else 6 mai	m tail som the
	\$	
	v EPJ [3] = man (v[1-1][3), v[i	-13E1-WELTS - PLISS.
		- J E 23 11 12 13 1
-	3 - The second second second	¥2
	3	TOURS I
	Print + 1" The autout Par sans	The second bill be a part
	Print + 1" The output is: \n");	traue trusts
	for (1=0; 1<=n; 1++)	2
		Harmal
	for 1 =0; j = m; j++)	in the same of the
	\$	- Land
	Print + (""/d\t", vei3 cs);	Output
	ξ: 11.9Fi	Corer Upe. no c
	Print ("\n\n") dans to the	fater the con
	3	S.
	Print f 1" In The optimal colution	nis /d", v[n][m]);
	print + ("In Solution vector is	: ("):
	for 11=n; 1>=1; 1) 119+1 drag	170 til 210 1919
	1 'C	and a respect of
	([m][1-1] v = [m][1]v] + i	70
	S	- may have the Ohear
		i epter Knap sock
	m=m-WE1);	21 to the gulf
	2	a a a a a
	0100	A A A A
	20 20 2	01 01 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	3-10	The same of the sa
CS_	Starmed with CamScarmer	01 00 00
	9 35 19 00	it to storitge all

1	Class Date Page 2
-	3
	for (i=1; i = n; i++)
	S = CONCITUO-11-11
	Print + 1" > d \ t", n [i]);
	3 - STICL-ITY CARCAS
	int max (int n, inty)
	-
	-(E) 19 FUM >43-13CL-13 v. C(1CL-13 v. K pm : C) 1637
	\$
	return n;
	3
	S :("ar . 2" tugtus out " 1 1 mill
	returny:
	3 (rrijai=2; 0=11 vok
	Enter the no of them ?
	Enter the no of item: 3
	Enter the weight of each item a/ this
	6 The application of the
	101.11/ 201.11
	enter profit of each 1 tem
	20
	30 - (-n/[1-1] v = ([m] 1/[v]) 1/1
	enter knap sack capacty:5
	11 110 011 10 1 10
	0 0 0 0 0 0
	0 0 10 10 10
	0 0 10 10 20 20
	0 0 10 10 20 20 1000
·S 0	The apthroal column 1500
-1-3	The solution vector is 0 1 0.