

```

#include //include void selectionsort(); void radixsort(); int main() { int choice; //clrscr(); do {
printf("\n\t\tProgram for Selection Sort and Radix Sort \n"); printf("\n\tMain Menu: \n1.Selection
Sort\n2.Radix Sort\n3.Exit"); printf("\nSelect menu : "); scanf("%d",&choice); switch(choice) {
case 1: selectionsort(); break; case 2: radixsort(); break; case 3: printf("\nExiting the program");
break; default: printf("Invalid menu item selected."); } }while(choice != 3); return 0; } void
selectionsort() { int array[100],i,j,k,n,smallest,position,temp; //clrscr(); printf("Enter number of
elements for selection sort\n"); scanf("%d", &n); printf("Enter %d integers\n", n); for (i = 0; i < n;
i++) { scanf("%d", &array[i]); } for (i = 0 ; i <(n - 1); i++) //It is for n-1 passes { smallest = array[i];
position = i; for(j=(i+1); j < n) { NOP++;//This Loop is used to decide Max. Number of Passes
large/=10; } for(pass = 0; pass < NOP; pass++) { for(i = 0; i < 10; i++) { bucket_count[i] = 0; }
for(i = 0; i < n; i++)//The Outpur of this loop is find count of respective Bucket //& Place the
elements in that Bucket { remainder = (a[i] / divisor) % 10; bucket[remainder]
[bucket_count[remainder]] = a[i]; printf("\n bucket[%d][%d] =
%d",remainder,bucket_count[remainder],a[i]); // getch(); bucket_count[remainder] += 1; } i = 0;
for(k = 0; k < 10; k++) { for(j = 0; j < bucket_count[k]; j++) { a[i] = bucket[k][j]; //Collect the Output
of Pass 1 in Array a[] i++; } } divisor *= 10; printf("\nOutput of this Pass:"); for(i = 0; i < n; i++)
printf("%d ",a[i]);//Show the Output of Pass printf("\n"); } printf("The sorted elements are :: "); for(i
= 0; i < n; i++) printf("%d ",a[i]); printf("\n"); //getch(); }

```