// Radix sort

#include <stdio.h>

int count\_sort(int arr[], int n, int exp)

{

    int countarr[10] = {0};

    for (int i = 0; i < n; i++)

        countarr[(arr[i] / exp) % 10]++;

    for (int i = 1; i < 10; i++)

        countarr[i] = countarr[i - 1] + countarr[i];

    int output[n];

    for (int i = n - 1; i >= 0; i--)

    {

        output[countarr[(arr[i] / exp) % 10] - 1] = arr[i];

        countarr[(arr[i] / exp) % 10]--;

    }

    for (int i = 0; i < n; i++)

        arr[i] = output[i];

}

int radix\_sort(int arr[], int n)

{

    int max = 0;

    for (int i = 0; i < n; i++)

    {

        if (max < arr[i])

            max = arr[i];

    }

    for (int exp = 1; max / exp > 0; exp \*= 10)

        count\_sort(arr, n, exp);

}

int main()

{

    int n;

    printf("Enter the Number of Elements\n");

    scanf("%d", &n);

    int input[n];

    printf("Enter Elements\n");

    for (int i = 0; i < n; i++)

        scanf("%d", &input[i]);

    printf("The sorted array is \n");

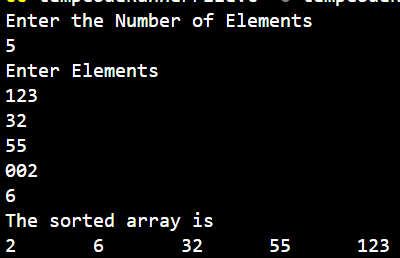
    radix\_sort(input, n);

    for (int i = 0; i < n; i++)

        printf("%d\t", input[i]);

}

Output:



#include <stdio.h>

#include <stdlib.h>

struct node

{

    int data;

    struct node \*next;

};

struct node \*arr[10];

void clear()

{

    for (int i = 0; i < 10; i++)

    {

        arr[i] = NULL;

    }

}

struct node \*create(struct node \*head, int dta)

{

    struct node \*newnode = (struct node \*)malloc(sizeof(struct node));

    newnode->data = dta;

    newnode->next = NULL;

    if (head == NULL)

    {

        head = newnode;

    }

    else

    {

        struct node \*temp = head;

        while (temp->next != NULL)

        {

            temp = temp->next;

        }

        temp->next = newnode;

    }

    return head;

}

struct node \*Create\_Node()

{

    struct node \*tmp;

    tmp = (struct node \*)malloc(sizeof(struct node));

    printf("Enter Data : ");

    scanf("%d",&tmp->data);

    tmp->next = NULL;

    return tmp;

}

struct node \*CreateLL(int n)

{

    struct node \*head = Create\_Node();

    struct node \*tmp = head;

    struct node \*NewNode;

    for(int i=1;i<n;i++)

    {

        NewNode = Create\_Node();

        tmp->next = NewNode;

        tmp = tmp->next;

    }

    return head;

}

void sort(struct node \*head, int exp)

{

    struct node \*temp = head;

    while (temp != NULL)

    {

        int i = temp->data;

        int place = (i / exp) % 10;

        arr[place] = create(arr[place], i);

        temp = temp->next;

    }

}

struct node \*joining(struct node \*new)

{

    for (int i = 0; i < 10; i++)

    {

        if (arr[i] == NULL)

        {

            continue;

        }

        struct node \*temp = arr[i];

        while (temp != NULL)

        {

            new = create(new, temp->data);

            temp = temp->next;

        }

    }

    return new;

}

void print(struct node \*head)

{

    struct node \*temp = head;

    while (temp != NULL)

    {

        printf("%d ", temp->data);

        temp = temp->next;

    }

    printf("\n");

}

void main()

{

    int n;

    clear();

    struct node \*head = NULL;

    struct node \*new = NULL;

    printf("Enter Number of Elements\n");

    scanf("%d",&n);

    head = CreateLL(n);

    print(head);

    printf("\n");

    for (int i = 1; i <= 100; i \*= 10)

    {

        sort(head, i);

        new = joining(new);

        print(new);

        clear();

        head = new;

        new = NULL;

    }

}

Output:

