

NAME: SADIKU DAVID JESUFERANMI

MATRIC no: 18/SCI01/082

### Question

40. Write a program in C that count from 20 to 1000 varying the control variable in the steps of 3 and printing it in form of a table. Using a) While statement and b) For statement. (Note, write the program separately). 41. Write a program that counts from 1 to 3000, prints the value: (a) vertically (b) horizontally separated with comma. Using Do-While Loop statement. 42. Write a program that reads an integer and displays its multiplication table. For example, if the user enters 5, the program should output:  $1*5 = 5$ ,  $2*5 = 10$ , ...,  $10*5 = 50$ . The program should force the user to enter an integer within [1, 10]. 43. Write a program that reads two integers and displays the sum of the integers between them. For example, if the user enters 3 and 8, the program should display 22 because  $4+5+6+7 = 22$ . The program should check which one of the two input numbers is the greater and act accordingly. 44. Write a program that reads an integer and displays a message to indicate whether it is a prime number or not. It is reminded that a prime number is any integer greater than 1 with no divisor other than 1 and itself. 45. Write a program that displays the multiplication table from 1 to 10. 46. Write a program that declares an array of five elements and uses a For loop to assign the values 1, 2, 3, 4, and 5 to them. Then, the program should display the array's elements in reverse order. 47. Write a program that reads two integers, stores them in two variables, declares two pointers to them, and displays the memory addresses of both variables, the content of both pointers, as well as their memory addresses. 48. Write a program that uses a pointer to read a float number and display its absolute value. 49. Write a C program for these given strings: 1) are You a NIGERIAN 2) I Come from Niger NIGERIA. a. Compare the two strings b. Compare at most n characters the two strings and c. Convert the second string to upper case. 50. Consider these given strings: Str1: My name is Samuel Samuel? Str2: I want to be a good programmer. a. Use strcat to join the two strings together. b. Use strset to replace specified characters with the string in str1. c. Convert string 1 to lower case. 51. Write a program that takes 10 words from user and sorts elements in lexicographical order/dictionary order. (Hint: two dimensional string, strcpy and strcmp is used) 52. Write a C program that concatenates two strings using strncat()

### SOLUTION

```
40(a) #include <stdio.h>
```

```
int main () {
```

```
    int num = 20;
```

```
    while (num<= 1000) {
```

```
        printf("%d ",num);
```

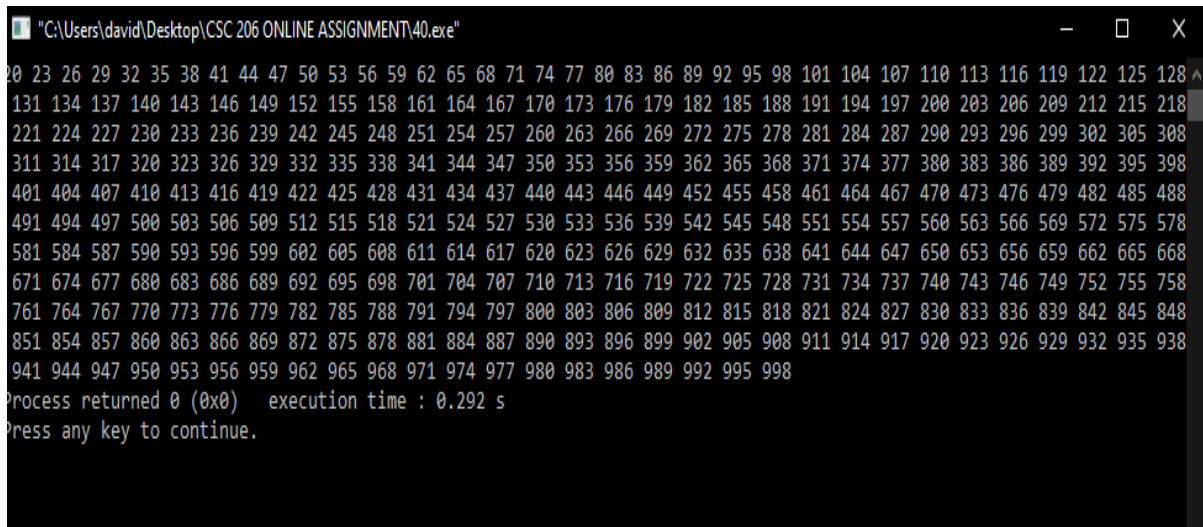
```
        num= num +3;
```

```
    }
```

```
}
```

```
b. #include<stdio.h>
```

```
int main(){  
    int num;  
    for(num= 20; num <= 1000; num=num+3){  
        printf("%d ", num);  
    }  
}
```



```
"C:\Users\david\Desktop\CSC 206 ONLINE ASSIGNMENT\40.exe"  
20 23 26 29 32 35 38 41 44 47 50 53 56 59 62 65 68 71 74 77 80 83 86 89 92 95 98 101 104 107 110 113 116 119 122 125 128  
131 134 137 140 143 146 149 152 155 158 161 164 167 170 173 176 179 182 185 188 191 194 197 200 203 206 209 212 215 218  
221 224 227 230 233 236 239 242 245 248 251 254 257 260 263 266 269 272 275 278 281 284 287 290 293 296 299 302 305 308  
311 314 317 320 323 326 329 332 335 338 341 344 347 350 353 356 359 362 365 368 371 374 377 380 383 386 389 392 395 398  
401 404 407 410 413 416 419 422 425 428 431 434 437 440 443 446 449 452 455 458 461 464 467 470 473 476 479 482 485 488  
491 494 497 500 503 506 509 512 515 518 521 524 527 530 533 536 539 542 545 548 551 554 557 560 563 566 569 572 575 578  
581 584 587 590 593 596 599 602 605 608 611 614 617 620 623 626 629 632 635 638 641 644 647 650 653 656 659 662 665 668  
671 674 677 680 683 686 689 692 695 698 701 704 707 710 713 716 719 722 725 728 731 734 737 740 743 746 749 752 755 758  
761 764 767 770 773 776 779 782 785 788 791 794 797 800 803 806 809 812 815 818 821 824 827 830 833 836 839 842 845 848  
851 854 857 860 863 866 869 872 875 878 881 884 887 890 893 896 899 902 905 908 911 914 917 920 923 926 929 932 935 938  
941 944 947 950 953 956 959 962 965 968 971 974 977 980 983 986 989 992 995 998  
Process returned 0 (0x0) execution time : 0.292 s  
Press any key to continue.
```

41a. **Vertically**

```
#include <stdio.h>  
  
int main () {  
    int num = 1;  
  
    do {  
        printf(" %d\n", num);  
        num++;  
    }  
  
    while( num <= 3000 );  
  
    return 0;  
}
```

"C:\Users\david\Desktop\CSC 206 ONLINE ASSIGNMENT\41a.exe"

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30
```

#### 41b. Horizontally separated with comma

```
#include <stdio.h>
```

```
int main () {
```

```
    int num = 1;
```

```
    do {
```

```
        printf("%d,", num);
```

```
        num++;
```

```
    }while( num <= 3000 );
```

```
    return 0;
```

```
}
```

```
"C:\Users\david\Desktop\CSC 206 ONLINE ASSIGNMENT\41b.exe"
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,
44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83,
84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117,
118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147,
148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177,
178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207,
208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237,
238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267,
268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297,
298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327,
328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357,
358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387,
388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417,
418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447,
448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477,
478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507,
508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537,
538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567,
568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597,
598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627,
628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657,
658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687,
688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717,
718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747,
748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777,
778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807,
808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837,
838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867,
868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897,
898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928,
```

42. #include <stdio.h>

```
int main() {  
  
    int num, i = 1;  
  
    printf("Enter any Number: ");  
  
    scanf("%d", &num);  
  
    printf("Multiplication table of %d:\n ", num);  
  
    while (i <= 10) {  
  
        printf("%d x %d = %d\n", num, i, num * i);  
  
        i++;  
  
    }  
  
    return 0;  
  
}
```

```
"C:\Users\david\Desktop\CSC 206 ONLINE ASSIGNMENT\42.exe"
Enter any Number: 45
Multiplication table of 45:
 45 x 1 = 45
45 x 2 = 90
45 x 3 = 135
45 x 4 = 180
45 x 5 = 225
45 x 6 = 270
45 x 7 = 315
45 x 8 = 360
45 x 9 = 405
45 x 10 = 450

Process returned 0 (0x0)   execution time : 3.833 s
Press any key to continue.
```

```
43. #include <stdio.h>
```

```
int main()
```

```
{
```

```
    int a = 0;
```

```
    int b = 0;
```

```
    int c=0;
```

```
    int total_sum = 0;
```

```
    printf("input the first number : ");
```

```
    scanf("%d", &a);
```

```
    printf("input the second number : ");
```

```
    scanf("%d", &b);
```

```
    while (a > b) {
```

```
        printf("The second number should be bigger than the first one.\n");
```

```
        printf("Type the first number : ");
```

```
        scanf("%d", &a);
```

```
printf("Type the second number : ");
scanf("%d", &b);
}
while (a < b) {
    total_sum += c;
    ++a;
}
printf("Result : %d\n", total_sum);
}
```

44. #include <stdio.h>

```
int main() {
    int n, i, isPrime = 0;
    printf("Enter a positive integer: ");
    scanf("%d", &n);

    for (i = 2; i <= n / 2; ++i) {

        // condition for non-prime
        if (n % i == 0) {
            isPrime = 1;
            break;
        }
    }

    if (n == 1) {
```

```

    printf("1 is neither prime nor composite.");
}
else {
    if (isPrime == 0)
        printf("%d is a prime number.", n);
    else
        printf("%d is not a prime number.", n);

    printf("It is reminded that a prime number is any integer greater than 1 with no divisor other
than 1 and itself. ");
}

return 0;
}

```

```

C:\Users\david\Desktop\CSC 206 ONLINE ASSIGNMENT\44.exe
Enter a positive integer: 7
7 is a prime number.
A prime number is any integer greater than 1 with no divisor other than 1 and itself.
Process returned 0 (0x0) execution time : 2.548 s
Press any key to continue.

```

```

45. #include <stdio.h>

void main()
{
    int j,i,n;

    printf("Input upto the table number starting from 1 : ");

    scanf("%d",&n);

```

```

printf("Multiplication table from 1 to %d \n",n);

for(i=1;i<=10;i++)
{
    for(j=1;j<=n;j++)
    {
        if (j<=n-1)
            printf("%dx%d= %d ",j,i,i*j);

        else
            printf("%dx%d= %d",j,i,i*j);

    }

    printf("\n");
}
}

```

```

"C:\Users\David\Desktop\CSC 206 ONLINE ASSIGNMENT\45.exe"
Input upto the table number starting from 1 : 10
Multiplication table from 1 to 10
1x1= 1 2x1= 2 3x1= 3 4x1= 4 5x1= 5 6x1= 6 7x1= 7 8x1= 8 9x1= 9 10x1= 10
1x2= 2 2x2= 4 3x2= 6 4x2= 8 5x2= 10 6x2= 12 7x2= 14 8x2= 16 9x2= 18 10x2= 20
1x3= 3 2x3= 6 3x3= 9 4x3= 12 5x3= 15 6x3= 18 7x3= 21 8x3= 24 9x3= 27 10x3= 30
1x4= 4 2x4= 8 3x4= 12 4x4= 16 5x4= 20 6x4= 24 7x4= 28 8x4= 32 9x4= 36 10x4= 40
1x5= 5 2x5= 10 3x5= 15 4x5= 20 5x5= 25 6x5= 30 7x5= 35 8x5= 40 9x5= 45 10x5= 50
1x6= 6 2x6= 12 3x6= 18 4x6= 24 5x6= 30 6x6= 36 7x6= 42 8x6= 48 9x6= 54 10x6= 60
1x7= 7 2x7= 14 3x7= 21 4x7= 28 5x7= 35 6x7= 42 7x7= 49 8x7= 56 9x7= 63 10x7= 70
1x8= 8 2x8= 16 3x8= 24 4x8= 32 5x8= 40 6x8= 48 7x8= 56 8x8= 64 9x8= 72 10x8= 80
1x9= 9 2x9= 18 3x9= 27 4x9= 36 5x9= 45 6x9= 54 7x9= 63 8x9= 72 9x9= 81 10x9= 90
1x10= 10 2x10= 20 3x10= 30 4x10= 40 5x10= 50 6x10= 60 7x10= 70 8x10= 80 9x10= 90 10x10= 100

Process returned 10 (0xA)   execution time : 0.895 s
Press any key to continue.

```

46. #include <stdio.h>

int main()



```
{  
    //Initialize array  
    int arr[] = {1, 2, 3, 4, 5};  
  
    //Calculate length of array arr  
    int length = sizeof(arr)/sizeof(arr[0]);  
  
    printf("Original array: \n");  
    for (int i = 0; i < length; i++) {  
        printf("%d ", arr[i]);  
    }  
  
    printf("\n");  
  
    printf("Array in reverse order: \n");  
    //Loop through the array in reverse order  
    for (int i = length-1; i >= 0; i--) {  
        printf("%d ", arr[i]);  
    }  
    return 0;  
}
```

```
"C:\Users\david\Desktop\CSC 206 ONLINE ASSIGNMENT\46.exe"
Original array:
1 2 3 4 5
Array in reverse order:
5 4 3 2 1
Process returned 0 (0x0)   execution time : 0.079 s
Press any key to continue.
```

```
47. #include <stdio.h>
```

```
int main()
```

```
{
```

```
    int var ;
```

```
    int *p;
```

```
    p= &var;
```

```
    printf ( "\nValue of var is: ");
```

```
    scanf("%d",&var);
```

```
    printf ( "\nValue of var is: ");
```

```
    scanf("%d",&*p);
```

```
    printf ( "\nAddress of var is: %p", &var);
```

```
    printf ( "\nAddress of var is: %p", p);
```

```
    /* Note I have used %p for p's value as it represents an address*/
```

```
    printf( "\nValue of pointer p is: %p", p);
```

```
    printf ( "\nAddress of pointer p is: %p", &p);
```

```
return 0;
}
```

```
Select "C:\Users\david\Desktop\CSC 206 ONLINE ASSIGNMENT\47.exe"

Value of var is: 34
Value of var is: 23
Address of var is: 0060FEFC
Address of var is: 0060FEFC
Value of pointer p is: 0060FEFC
Address of pointer p is: 0060FEF8
Process returned 0 (0x0) execution time : 6.872 s
Press any key to continue.
```

48.

```
#include<stdio.h>
```

```
#include<string.h>
```

```
int main() {
```

```
    char str1[]= "are You a NIGERIAN";
```

```
    char str2[]= "I Come from Niger NIGERIA. ";
```

```
    if (strcmp(str1,str2)==0)
```

```
    {
```

```
        printf("\n\n(a) String 1 and String 2 are equal");
```

```
    }else{
```

```

        printf("\n\n(a(ii)) String 1 and String 2 are different");
    }
    if (strncmp(str1,str2,19)==0)
    {
        printf("\n\n(b) String 1 and String 2 are equal");
    }else{
        printf("\n\n(b(ii)) String 1 and String 2 are different");
    }

    printf("\n\n(c) The uppercase of string 2 is :%s",strupr(str2));
}

```

```

"C:\Users\david\Desktop\CSC 206 ONLINE ASSIGNMENT\49.exe"

(a) String 1 and String 2 are different
(b) String 1 and String 2 are different
(c) The uppercase of string 2 is :I COME FROM NIGER NIGERIA.
Process returned 0 (0x0)   execution time : 0.131 s
Press any key to continue.

```

```
50. #include<stdio.h>
```

```
#include<string.h>
```

```
int main() {
```

```
    char str1[]= " : My name is Samuel ";
```

```
    char str2[]= "I want to be a good programmer";
```

```

strcat(str1,str2);

printf("(i) Concatenation = %s ", str1);

printf("\n\n(b) Original string is : %s", str1);

printf(" \nstrset() : %s",strset(str1,'#'));

printf("\n\n(c) String 1 in lowercase: %s ", strlwr(str1));

return 0;
}

```

```

"C:\Users\david\Desktop\CSC 206 ONLINE ASSIGNMENT\50.exe"
(i) Concatenation = : My name is Samuel I want to be a good programmer
(b) Original string is : : My name is Samuel I want to be a good programmer
strset() : #####
(c) String 1 in lowercase: #####
Process returned -1073741819 (0xC0000005)   execution time : 11.173 s
Press any key to continue.

```

51. #include<stdio.h>

```
#include<string.h>

int main()
{
    char str[10][50],temp[50];
    int i,j;

    printf("Enter 10 Words:\n");

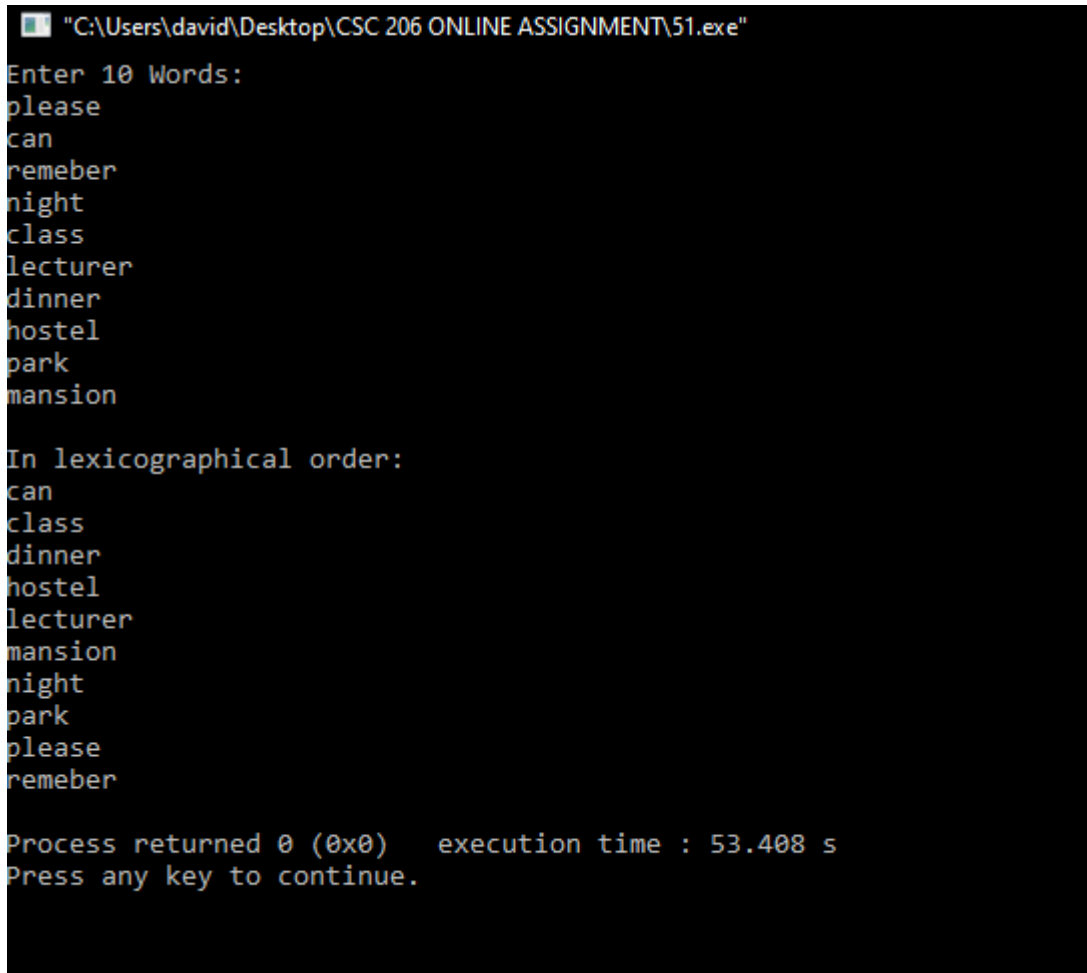
    for(i=0;i<10;i++)
        scanf("%s[^\n]",str[i]);

    for(i=0;i<9;i++)
    {
        for(j=i+1;j<10;j++)
        {
            if(strcmp(str[i],str[j])>0)
            {
                strcpy(temp,str[i]);
                strcpy(str[i],str[j]);
                strcpy(str[j],temp);
            }
        }
    }

    printf("\n\nIn lexicographical order: \n");

    for(i=0;i<10;i++)
        puts(str[i]);
}
```

```
return 0;
}
```



```
"C:\Users\David\Desktop\CSC 206 ONLINE ASSIGNMENT\51.exe"
Enter 10 Words:
please
can
remeber
night
class
lecturer
dinner
hostel
park
mansion

In lexicographical order:
can
class
dinner
hostel
lecturer
mansion
night
park
please
remeber

Process returned 0 (0x0)   execution time : 53.408 s
Press any key to continue.
```

```
52. #include <stdio.h>
```

```
#include <string.h>
```

```
int main()
```

```
{
```

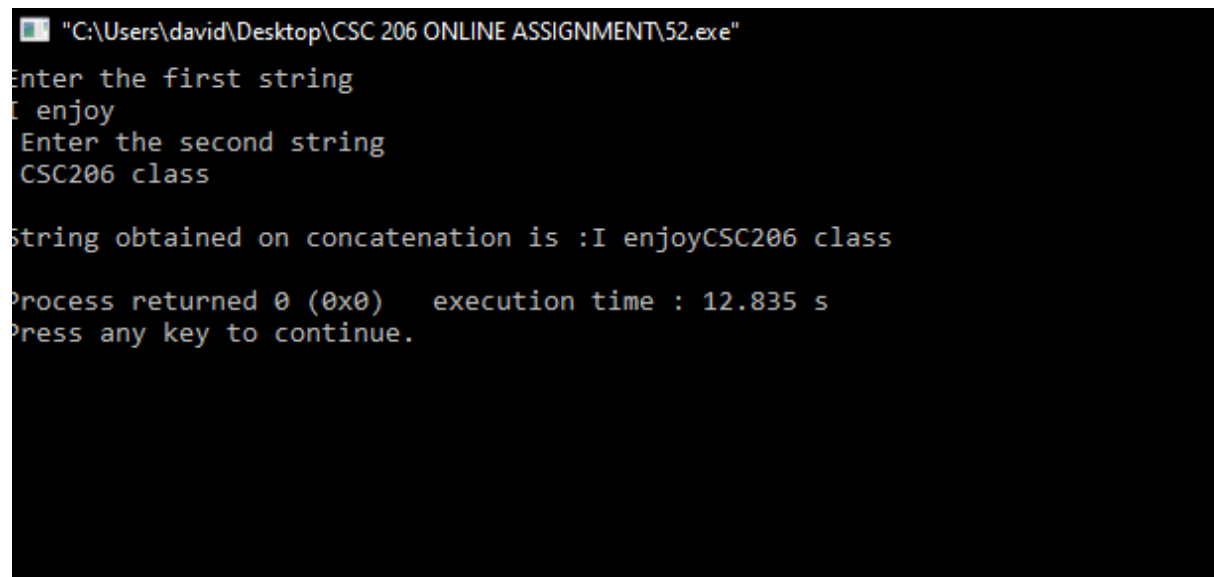
```
    char a[100], b[100];
```

```
    printf("Enter the first string\n");
```

```
    gets(a);
```

```
    printf("Enter the second string\n ");
```

```
gets(b);  
strcat(a,b);  
printf("String obtained on concatenation is %s\n",a);  
  
return 0;  
  
}
```



```
"C:\Users\david\Desktop\CSC 206 ONLINE ASSIGNMENT\52.exe"  
Enter the first string  
I enjoy  
Enter the second string  
CSC206 class  
  
String obtained on concatenation is :I enjoyCSC206 class  
  
Process returned 0 (0x0)   execution time : 12.835 s  
Press any key to continue.
```