1 October Lab Programs

# WAP to check whether the two strings are anagrams or not

1. // WAP to check whether the two strings are anagrams or not
2. #include <stdio.h>
3. int stringlen(const char \*s){
4. int c = 0;
5. while (\*s != '\0'){
6. c++;
7. s++;
8. }
9. return c;
10. }
11. int isAnagram(const char \*str1, const char \*str2){
12. int len1 = stringlen(str1);
13. int len2 = stringlen(str2);
14. if (len1 != len2){
15. return 0;
16. }
17. int letterfreq1[512] = {0};
18. int letterfreq2[512] = {0};
19. int i;
20. // counting the frequency of each letter,
21. // and storing it at it's ascii value as the index for both str1 and str2
22. for (i=0;i<len1;i++){
23. letterfreq1[str1[i]]++;
24. }
25. for (i=0;i<len2;i++){
26. letterfreq2[str2[i]]++;
27. }
28. for (i=0;i<512;i++){
29. if (letterfreq1[i] != letterfreq2[i]){
30. return 0;
31. }
32. }
33. return 1;
34. }
35. int main(){
36. const char \*stra = "listen",\*strb = "silent",\*strc = "hello",\*strd = "world";
37. if (isAnagram(stra,strb)){
38. printf("\"%s\" and \"%s\" are anagrams.\n",stra,strb);
39. } else{
40. printf("\"%s\" and \"%s\" are not anagrams.\n",stra,strb);
41. }
42. if (isAnagram(strc,strd)){
43. printf("\"%s\" and \"%s\" are anagrams\n", strc, strd);
44. } else{
45. printf("\"%s\" and \"%s\" are not anagrams\n",strc,strd);
46. }
47. return 0;
48. }

# WAP to find the second largest element in an array by passing the array to a function

1. // WAP to find the second largest element in an array by passing the array to a function
2. #include <stdio.h>
3. int second\_largest(int arr[],int size){
4. if (size<2){
5. printf("Array must have at least 2 elements\n");
6. return -1;
7. }
8. int l,sec\_l;
9. if (arr[0]>arr[1]){
10. l=arr[0];
11. sec\_l=arr[1];
12. }
13. else{
14. l=arr[1];
15. sec\_l=arr[0];
16. }
17. int i;
18. for (i=0;i<size;i++){
19. if (arr[i]>l){
20. sec\_l=l;
21. l=arr[i];
22. }
23. else if ((arr[i]>sec\_l) && (arr[i]<l)){
24. sec\_l = arr[i];
25. }
26. }
27. return sec\_l;
28. }
29. int main(){
30. int n,i;
31. printf("Enter the number of elements: ");
32. scanf("%d",&n);
33. int arr[n];
34. for (i=0;i<n;i++){
35. printf("Enter element %d: ",i+1);
36. scanf("%d",&arr[i]);
37. }
38. int sec\_large = second\_largest(arr,n);
39. if (sec\_large != -1){
40. printf("The second largest element is %d\n",sec\_large);
41. }
42. else{
43. printf("%d",sec\_large);
44. }
45. return 0;
46. }

# WAP to find sum of digits using recursion

1. // WAP to find sum of digits using recursion
2. #include <stdio.h>
3. int sumdigrec(int n){
4. if (n==0){
5. return 0;
6. }
7. else{
8. return (n%10) + sumdigrec(n/10);
9. }
10. }
11. int main(){
12. int num;
13. printf("Enter a number: ");
14. scanf("%d",&num);
15. int sum = sumdigrec(num);
16. printf("Sum of digits of %d is: %d",num,sum);
17. return 0;
18. }

# WAP to make function to swap two numbers using call by reference

1. // WAP to make function to swap two numbers using call by reference
2. #include <stdio.h>
3. void swapnum(int \*a, int \*b) {
4. int temp = \*a;
5. \*a = \*b;
6. \*b = temp;
7. }
8. int main(){
9. int a,b;
10. printf("Enter the value of a: "); scanf("%d",&a);
11. printf("Enter the value of b: "); scanf("%d",&b);
12. swapnum(&a,&b);
13. printf("\nnew value of a: %d",a);
14. printf("\nnew value of b: %d",b);
15. }

# Write a function to sort an array

1. // Write a function to sort an array
2. #include <stdio.h>
3. int arrsort(int arr[],int size){
4. int temp,i;
5. for (i=0;i<size;i++){
6. for (int j=i+1;j<size;j++){
7. if (arr[j]<arr[i]){
8. temp = arr[i];
9. arr[i] = arr[j];
10. arr[j] = temp;
11. }
12. }
13. }
14. }
15. int main(){
16. int n,i;
17. printf("Enter the number of elements: "); scanf("%d",&n);
18. int arr[n];
19. for (i=0;i<n;i++){
20. printf("Enter element %d:",i+1); scanf("%d",&arr[i]);
21. }
22. printf("Original Array: ");
23. for (i=0;i<n;i++){
24. if (i==n-1){
25. printf("%d",arr[i]);
26. }
27. else{
28. printf("%d, ",arr[i]);
29. }
30. }
31. arrsort(arr,n);
32. printf("\nSorted Array: ");
33. for (i=0;i<n;i++){
34. if (i==n-1){
35. printf("%d",arr[i]);
36. }
37. else{
38. printf("%d, ",arr[i]);
39. }
40. }
41. }

# WAP to define a structure named 'student' with roll no., name and marks and write functions to input and display student details

1. // WAP to define a structure named 'student' with roll no., name and marks and write
2. // functions to input and display student details
3. #include <stdio.h>
4. #define MAX\_SIZE 1000
5. struct student{
6. int rollno;
7. char name[100];
8. float marks;
9. }; struct student stud\_data[MAX\_SIZE];
10. int structinput(){
11. int n,i;
12. printf("Enter the number of student records: ");
13. scanf("%d",&n);
14. for (i=0;i<n;i++){
15. printf("\n\nEnter the details for student %d:\n",i+1);
16. printf("Enter student's roll no.: "); scanf(" %d",&stud\_data[i].rollno);
17. printf("Enter student's name: "); scanf(" %s",&stud\_data[i].name);
18. printf("Enter student's marks (in decimal): "); scanf(" %f",&stud\_data[i].marks);
19. }
20. return n;
21. }
22. int main(){
23. int n = structinput();
24. int i;
25. for (i=0;i<n;i++){
26. printf("\nroll: %d, name:%s, marks:%.2f",stud\_data[i].rollno,
27. stud\_data[i].name,stud\_data[i].marks);
28. }
29. return 0;
30. }

# WAP to create a structure employee (ID, Name, Basic Pay, DA %, HRA %, (to calc) Gross Salary).and write a function to calculate salary and display details.

1. // WAP to create a structure employee to input (ID, Name, Basic Pay, DA %, HRA %) and to
2. // calc and print - (Gross Salary).
3. // and write a function to calculate salary and display details.
4. #include <stdio.h>
5. #define MAX\_SIZE 1000
6. struct employee{
7. int ID;
8. char name[100];
9. float basic\_pay;
10. float DA;
11. float HRA;
12. float gross\_salary;
13. }; struct employee emp\_data[MAX\_SIZE];
14. int empInput(){
15. int n,i;
16. printf("Enter the number of employee record to add: "); scanf("%d",&n);
17. for (i=0;i<n;i++){
18. printf("\nEnter the details for employee %d: \n",i+1);
19. printf("Enter the employee's ID: "); scanf(" %d",&emp\_data[i].ID);
20. printf("Enter the employee's Name: "); scanf(" %s",emp\_data[i].name);
21. printf("Enter the employees Basic Pay: "); scanf(" %f",&emp\_data[i].basic\_pay);
22. printf("Enter the employees DA%%: "); scanf(" %f",&emp\_data[i].DA);
23. printf("Enter the employees HRA%%: "); scanf(" %f",&emp\_data[i].HRA);
24. }
25. return n;
26. }
27. int grossCalc(){
28. int n = empInput(),i;
29. float HRA\_P,DA\_P,basic,gross\_amt;
31. for (i=0;i<n;i++){
32. basic = emp\_data[i].basic\_pay;
33. HRA\_P = emp\_data[i].HRA;
34. DA\_P = emp\_data[i].DA;
35. gross\_amt = basic + (basic\*(HRA\_P/100)) + (basic\*(DA\_P/100));
36. emp\_data[i].gross\_salary = gross\_amt;
37. }
38. return n;
39. }
40. int displayDetails(){
41. int n = grossCalc(),i;
43. printf("The employee data is: \n");
44. for (i=0;i<n;i++){
45. printf("\nID:%d, Name:%s,\t Basic Pay:%.2f, %%-DA:%.2f, %%-HRA:%.2f,
46. Gross Salary:%.2f",emp\_data[i].ID,emp\_data[i].name,
47. emp\_data[i].basic\_pay,emp\_data[i].DA,
48. emp\_data[i].HRA,emp\_data[i].gross\_salary);
49. }
50. }
51. int main(){
52. displayDetails();
53. return 0;
54. }