**Theory Exam 08 Answer Script**

| Question No. 01 |
| --- |
| Question: Make a function named make\_sum() which will take two floating values as parameters and return their sum |
| Answer:  float make\_sum(float a, float b)  {  return a + b;  } |

| Question No. 02 |
| --- |
| Question: Make a function named make\_average() which will take an array of integers and the size of that array and return the average of those values. |
| Answer:  double make\_average(int n, int a[])  {  int sum;  for (int i = 0; i < n; i++)  sum = sum + a[i];  double avg = (double)sum / (double)n;  return avg;  } |

| Question No. 03 |
| --- |
| Question: Make a function named count\_zero() which will take a binary string (Binary string is a string which is consist of only 0 and 1) as parameter and count how many 0’s are there in that string |
| Answer:  int count\_zero(char s[])  {  int sum = 0;  for (int i = 0; i < strlen(s); i++)  if (s[i] == '0')  sum++;  return sum;  } |

| Question No. 04 |
| --- |
| Question: Make a function named foo() which prints “foo\n” and a function named bar() which prints “bar\n”. Call function foo() in the main() function and call function bar() in the foo() function after printing. What will be the output? |
| Answer:  void bar()  {  printf("bar\n");  }  void foo()  {  printf("foo\n");  bar();  }  int main()  {  foo();  return 0;  }  The output will be:  foo  bar |

| Question No. 05 |
| --- |
| Question: Make a function named check\_array() which will take an array of integers and the size of that array N. It will return a boolean type whether this array has all values from 1 to N or not. |
| Answer:  bool check\_array(int n, int a[])  {  int b[n + 1];  for (int i = 0; i <= n; i++)  b[i] = 0;  for (int i = 0; i < n; i++)  b[a[i]]++;  for (int i = 1; i <= n; i++)  if (b[i] == 0)  return false;  return true;  } |

| Question No. 06 |
| --- |
| Question: Make a pointer variable P which points to an integer variable. Make another pointer variable Q which points to the pointer P. Now make another pointer variable R which points to the pointer Q. Now change the value of that integer variable by accessing pointer R. |
| Answer:  #include <stdio.h>  int main() {  int a = 10;  int \*p = &a;  int \*\*q = &p;  int \*\*\*r = &q;  \*\*\*r= 30;  printf("%d", a);  } |

| Question No. 07 |
| --- |
| Question: Make a function named count\_swaps() which will take an array of integers and the size of that array. You need to tell how many swaps you need while implementing the selection sort that is shown in the module video and return that number of swaps from that function. |
| Answer:  int count\_swaps(int array[], int size) {  int count = 0;  for (int step = 0; step < size - 1; step++) {  int min\_idx = step;  for (int i = step + 1; i < size; i++)  if (array[i] < array[min\_idx]) {  min\_idx = i;  swap(&array[min\_idx], &array[step]);  count++;  }  }  return count;  } |

| Question No. 08 |
| --- |
| Question: Make a function named odd\_even() which takes an integer value and tells whether this value is even or odd. You need to do it in 4 ways:  i) Has return + Has parameter  ii) No return + Has parameter  iii) Has return + No parameter  iv) No return + No parameter |
| Answer:  i) Has return + Has parameter  int odd\_even(int a){  if (a % 2) return 1;  else return 0;  }  ii) No return + Has parameter  void odd\_even(int a){  if (a % 2) printf("%d is odd\n", a);  else printf("%d is even", a);  }  iii) Has return + No parameter  int odd\_even(){  int n;  scanf("%d", &n);  if (n % 2) return 1;  else return 0;  }  iv) No return + No parameter  void odd\_even(){  int n;  scanf("%d", &n);  if (n % 2) printf("%d is odd\n", n);  else printf("%d is even", n);  } |

| Question No. 09 |
| --- |
| Question: You know palindromes, right? Now make a function named check\_palindrome() which will take a string as a parameter and return the minimum number of characters you need to change so that the string can become palindrome. You can’t add or delete any character. For example: check\_palindrome(“abcdba”) will return 1 as you can change the character of index 2 to ‘d’ or character of index 3 to ‘c’ to make it palindrome. |
| Answer:  int check\_palindrome(char s[]) {  int sum = 0;  for (int i = 0; i < strlen(s) / 2; i++)  if (s[i] != s[strlen(s) - 1 - i])  sum++;  return sum;  } |

| Question No. 10 |
| --- |
| Question: Make a function named change\_array() which will take an integer array and size of that array. After that you will reverse that array and put that in a new array and print it in the main() function. You know that you can’t return an array normally, so you need to make that array in the main() function and pass that through the parameter. |
| Answer:  #include <stdio.h>  void change\_array(int n, int a[], int b[]) {  for (int i = 0; i < n; i++)  b[i] = a[n - 1 - i];  }  int main() {  int a[5] = {1, 2, 3, 4, 5};  int b[5];  change\_array(5, a, b);  for (int i = 0; i < 5; i++)  printf("%d ", b[i]);  return 0;  } |