SORU1:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define ARRAY\_SIZE 10

int max\_recursive(int arr[], int size) {

if (size == 1) {

return arr[0];

} else {

int max = max\_recursive(arr, size-1);

return arr[size-1] > max ? arr[size-1] : max;

}

}

int main() {

int arr[ARRAY\_SIZE];

int i;

// rastgele sayılarla dizi oluştur

srand(time(NULL));

for (i = 0; i < ARRAY\_SIZE; i++) {

arr[i] = rand() % 100;

}

// dizi elemanlarını ekrana yazdır

printf("Dizi: ");

for (i = 0; i < ARRAY\_SIZE; i++) {

printf("%d ", arr[i]);

}

// en büyük değeri bul ve ekrana yazdır

int max = max\_recursive(arr, ARRAY\_SIZE);

printf("\nEn buyuk deger: %d\n", max);

return 0;

}

SORU2:

#include <stdio.h>

#include <string.h>

void reverse\_string(char str[], int start, int end) {

if (start >= end) {

return;

} else {

char temp = str[start];

str[start] = str[end];

str[end] = temp;

reverse\_string(str, start+1, end-1);

}

}

int string\_length(char str[]) {

if (str[0] == '\0') {

return 0;

} else {

return 1 + string\_length(str+1);

}

}

int main() {

char str[100];

printf("Bir string girin: ");

fgets(str, 100, stdin);

// stringin uzunluğunu hesapla

int len = string\_length(str);

// stringi tersten yaz

reverse\_string(str, 0, len-1);

printf("Tersten yazilmis hali: %s\n", str);

printf("Uzunluk: %d\n", len);

return 0;

}

SORU3:

A)

#include <stdio.h>

int fibonacci\_recursive(int n) {

if (n <= 1) {

return n;

} else {

return fibonacci\_recursive(n-1) + fibonacci\_recursive(n-2);

}

}

int main() {

int n;

printf("Fibonacci serisi icin bir sayi girin: ");

scanf("%d", &n);

printf("%d. Fibonacci sayisi: %d\n", n, fibonacci\_recursive(n));

return 0;

}

B) #include <stdio.h>

int fibonacci\_iterative(int n) {

int a = 0, b = 1, i;

if (n == 0) {

return a;

}

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

int temp = b;

b = a + b;

a = temp;

}

return b;

}

int main() {

int n;

printf("Fibonacci serisi icin bir sayi girin: ");

scanf("%d", &n);

printf("%d. Fibonacci sayisi: %d\n", n, fibonacci\_iterative(n));

return 0;

}