Programming 4 beginners

Assignment 2



# Challenge 1

#include <stdio.h>

void print\_english(int n) {

if (1 <= n && n <= 9) {

switch (n) {

case 1: printf("one\n"); break;

case 2: printf("two\n"); break;

case 3: printf("three\n"); break;

case 4: printf("four\n"); break;

case 5: printf("five\n"); break;

case 6: printf("six\n"); break;

case 7: printf("seven\n"); break;

case 8: printf("eight\n"); break;

case 9: printf("nine\n"); break;

}

} else if (n > 9) {

if (n % 2 == 0) {

printf("even\n");

} else {

printf("odd\n");

}

}

}

int main() {

int a, b;

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

for (int n = a; n <= b; n++) {

print\_english(n);

}

return 0;

}

# Challenge 2

#include <stdio.h>

int main() {

int n;

scanf("%d", &n);

int size = 2 \* n - 1;

for (int i = 0; i < size; i++) {

for (int j = 0; j < size; j++) {

int min = i < j ? i : j;

min = min < size - i ? min : size - i - 1;

min = min < size - j - 1 ? min : size - j - 1;

printf("%d ", n - min);

}

printf("\n");

}

return 0;

}

# Challenge 3

#include <stdio.h>

#include <stdlib.h>

int main() {

int n, i, sum = 0;

int \*arr;

// Read number of elements

scanf("%d", &n);

// Dynamically allocate memory for n integers

arr = (int\*) malloc(n \* sizeof(int));

// Check if memory allocation was successful

if (arr == NULL) {

printf("Memory allocation failed\n");

return 1;

}

// Read array elements

for (i = 0; i < n; i++) {

scanf("%d", &arr[i]);

sum += arr[i]; // calculate sum while reading

}

// Print the sum

printf("%d\n", sum);

// Free the allocated memory

free(arr);

return 0;

}

# Challenge 4

#include <stdio.h>

int main() {

int n;

scanf("%d", &n);

int arr[n];

for (int i = 0; i < n; i++) {

scanf("%d", &arr[i]);

}

for (int i = 0; i < n / 2; i++) {

int temp = arr[i];

arr[i] = arr[n - 1 - i];

arr[n - 1 - i] = temp;

}

for (int i = 0; i < n; i++) {

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

}

return 0;

}

# Challenge 05:

Solution:

#include <stdio.h>

// Function to find maximum of two numbers

int max(int x, int y) {

if(x > y)

return x;

else

return y;

}

// Function to find maximum of four numbers

int max\_of\_four(int a, int b, int c, int d) {

return max(max(a, b), max(c, d));

}

int main() {

int a, b, c, d;

// Read four integers

scanf("%d %d %d %d", &a, &b, &c, &d);

// Find and print the maximum

int ans = max\_of\_four(a, b, c, d);

printf("%d", ans);

Return 0;

}

Challenge 6

#include <stdio.h>

int calculate\_nth\_term(int n, int a, int b, int c) {

if (n == 1) return a;

if (n == 2) return b;

if (n == 3) return c;

return calculate\_nth\_term(n-1, a, b, c) + calculate\_nth\_term(n-2, a, b, c) + calculate\_nth\_term(n-3, a, b, c);

}

int main() {

int n, a, b, c;

scanf("%d", &n);

scanf("%d %d %d", &a, &b, &c);

printf("%d\n", calculate\_nth\_term(n, a, b, c));

return 0;

}

# Challenge 7

#include <stdio.h>

void update(int \*a, int \*b) {

int sum = \*a + \*b;

int diff = (\*a > \*b) ? (\*a - \*b) : (\*b - \*a);

\*a = sum;

\*b = diff;

}

int main() {

int a, b;

scanf("%d\n%d", &a, &b);

update(&a, &b);

printf("%d\n%d", a, b);

return

Challenge 08

Solution:

#include <stdio.h>

int main() {

int n;

scanf("%d", &n);

int marks[n];

for(int i = 0; i < n; i++) {

scanf("%d", &marks[i]);

}

// Boys - even indices (0, 2, 4...)

for(int i = 0; i < n; i += 2) {

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

}

printf("\n");

// Girls - odd indices (1, 3, 5...)

for(int i = 1; i < n; i += 2) {

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

}

return 0;

}

# Challenge 09:

Solution:

#include <stdio.h>

#include <string.h>

int main() {

char s[1000];

// Read the entire sentence

fgets(s, sizeof(s), stdin);

// Remove newline character if present

s[strcspn(s, "\n")] = 0;

// Print each word on new line

for(int i = 0; s[i] != '\0'; i++) {

if(s[i] == ' ') {

printf("\n");

} else {

printf("%c", s[i]);

}

}

return 0;

}

# Challenge 10:

Solution:

#include <stdio.h>

int main() {

char s[1000];

int freq[10] = {0};

scanf("%s", s);

for(int i = 0; s[i] != '\0'; i++) {

if(s[i] >= '0' && s[i] <= '9') {

freq[s[i] - '0']++;

}

}

for(int i = 0; i < 10; i++) {

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

}

return 0;

}