**Technical test**

1. #**include** <iostream>

using namespace std;

int main() {

for (int i = 1; i <= 100; ++i) {

bool isFizz = (i % 3 == 0);

bool isBuzz = (i % 5 == 0);

if (isFizz && isBuzz) {

cout << "FizzBuzz" << endl;

} else if (isFizz) {

cout << "Fizz" << endl;

} else if (isBuzz) {

cout << "Buzz" << endl;

} else {

cout << i << endl;

}

}

return 0;

}

1. #**include** <iostream>

int fibonacci(int n) {

if (n == 0 || n == 1) {

return n;

} else {

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

}

}

int main() {

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

std::cout << fibonacci(i) << " ";

}

std::cout << std::endl;

return 0;

}

3. #**include** <iostream>

bool isPowerOfTwo(int n) {

return n > 0 && !(n & (n - 1)); // Check if only one bit is set

}

int main() {

int num;

std::cout << "Enter an integer: ";

std::cin >> num;

if (isPowerOfTwo(num)) {

std::cout << num << " is a power of two." << std::endl;

} else {

std::cout << num << " is not a power of two." << std::endl;

}

return 0;

}

4. #**include** <iostream>

#**include** <cctype>

#**include** <string>

std::string capitalizeWords(const std::string& str) {

std::string result;

bool capitalizeNext = true; // Flag to show if next letter should be capitalized

for (char c : str) {

if (std::isspace(c)) {

// Space encountered, reset flag and append space

capitalizeNext = true;

result += c;

} else if (capitalizeNext) {

// First letter of a word, convert to uppercase

result += std::toupper(c);

capitalizeNext = false;

} else {

// Lowercase letter, append directly

result += c;

}

}

return result;

}

int main() {

std::string input;

std::cout << "Enter a string: ";

std::getline(std::cin, input);

std::string capitalized = capitalizeWords(input);

std::cout << "Capitalized string: " << capitalized << std::endl;

return 0;

}

5. #**include** <iostream>

#**include** <limits>

int reverseInteger(int num) {

long long reversed = 0; // Use long long to handle large numbers

while (num != 0) {

int digit = num % 10; // Extract last digit

num /= 10; // Remove last digit

// Check for overflow using INT\_MAX/10 and INT\_MIN/10

if (reversed > INT\_MAX / 10 || (reversed == INT\_MAX / 10 && digit > 7)) {

return 0; // Overflow: return 0 to indicate error

} else if (reversed < INT\_MIN / 10 || (reversed == INT\_MIN / 10 && digit < -8)) {

return 0; // Overflow: return 0 to indicate error

}

reversed = reversed \* 10 + digit; // Add digit to reversed number

}

return reversed;

}

int main() {

int input;

std::cout << "Enter an integer: ";

std::cin >> input;

int reversed = reverseInteger(input);

if (reversed != 0) {

std::cout << "Reversed integer: " << reversed << std::endl;

} else {

std::cout << "Error: Integer overflow occurred." << std::endl;

}

return 0;

}

6. #**include** <iostream>

#**include** <string>

using namespace std;

int countVowels(const string& sentence) {

int vowelCount = 0;

string vowels = "aeiouAEIOU";

for (char c : sentence) {

// Check if character is a vowel using find

if (vowels.find(c) != string::npos) {

vowelCount++;

}

}

return vowelCount;

}

int main() {

string sentence;

cout << "Enter a sentence: ";

getline(cin, sentence);

int vowelCount = countVowels(sentence);

cout << "The sentence '" << sentence << "' has " << vowelCount << " vowels." << endl;

return 0;

}