

Załącznik nr 2 do Regulaminu Ogólnopolskiego
Branżowego Konkursu Umiejętności
w dziedzinie informatyki i programowania

Ogólnopolski Branżowy Konkurs Umiejętności w dziedzinie informatyki i programowania

Zakres wiedzy i umiejętności na poszczególnych etapach Konkursu

Konkurs przeznaczony jest dla uczniów kształcących się w zawodach technik informatyk oraz technik programista, umożliwiając rozwijanie ich kompetencji oraz zainteresowań w dziedzinie algorytmiki i programowania.

Celem konkursu jest:

- Rozwijanie zainteresowań informatyką i programowaniem.
- Kształtowanie umiejętności rozwiązywania problemów, wspieranie logicznego myślenia, kreatywności i umiejętności analitycznych poprzez różnorodne zadania.
- Wspieranie rozwoju kompetencji cyfrowych poprzez korzystanie z narzędzi cyfrowych w sposób bezpieczny i odpowiedzialny.
- Wzmacnianie umiejętności korzystania z podstawowych programów, algorytmów i kodowania.
- Odkrywanie i wspieranie talentów uczniów szczególnie zainteresowanych informatyką i programowaniem.
- Inspirowanie uczniów do eksplorowania dziedziny technologii, programowania i nowych mediów.
- Wyłanianie talentów informatycznych.
- Popularyzacja nowoczesnych technologii w tym e-learningowych.

Zakres wiedzy i umiejętności

Konkurs obejmuje i poszerza treści podstawy programowej kształcenia w zawodzie technik informatyk oraz technik programista. Zadania na kolejnych etapach będą różnicowane pod względem poziomu trudności. Zadania będą wymagały od uczestników umiejętności posługiwania się jednym z języków: C++, C#, Python. Wszystkie etapy konkursu przeprowadzone będą stacjonarnie. Na etapie szkolnym uczniowie rozwiązują zadania testowe pojedynczego wyboru oraz zadania krótkiej odpowiedzi (liczbowej) do wypełnienia na karcie odpowiedzi. Natomiast podczas półfinału i finału Konkursu uczestnicy będą mieli do dyspozycji samodzielne stanowisko komputerowe bez dostępu do Internetu, przy którym będą rozwiązywać zadania otwarte wymagające udzielenia odpowiedzi w postaci liczby lub tekstu. Odpowiedzi na zadania otwarte będą umieszczane przez ucznia w Autonomicznej Platformie Konkursowej (APK), jako

rozwiązania testu konkursowego. Po zakończeniu testu i zatwierdzeniu odpowiedzi uczestnik poszczególnego etapu otrzyma raport zawierający m. in. czas rozwiązywania zadań, liczbę punktów oraz uzyskany wynik procentowy.

Szczegółowy zakres wymaganej wiedzy i umiejętności:

I. ALGORYTMIKA

1. Formułowanie i analiza problemów

- a. formułowanie problemu w postaci specyfikacji, obejmującej:
 - i. opis danych wejściowych,
 - ii. opis oczekiwanych wyników;
- b. analizowanie problemu oraz identyfikowania jego elementów składowych;

2. Projektowanie i zapis algorytmów

- a. wyróżnianie kroków prowadzących do rozwiązania problemu;
- b. przedstawianie algorytmów w różnych postaciach, w szczególności w postaci:
 - i. opisu słownego,
 - ii. listy kroków,
 - iii. schematów blokowych,
 - iv. pseudokodu;

3. Stosowanie podstawowych algorytmów na liczbach, w szczególności:

- a. badanie podzielności liczb;
- b. wyodrębnianie cyfr liczby;
- c. wyznaczanie największego wspólnego dzielnika (algorytm Euklidesa);
- d. wyznaczanie wyrazów oraz sum prostych ciągów liczbowych, w szczególności ciągów definiowanych rekurencyjnie (np. ciąg Fibonacciego);
- e. wyznaczanie najmniejszej wspólnej wielokrotności;
- f. badanie pierwszości liczby;
- g. zamiana reprezentacji liczb pomiędzy systemami liczbowymi;
- h. wykonywanie działań na ułamkach z wykorzystaniem NWD i NWW;

4. Stosowanie algorytmów na tekstach, w szczególności:

- a. porównywanie tekstów;
- b. wyszukiwanie wzorca w tekście metodą naiwną;
- c. szyfrowanie tekstu, w szczególności szyfrem Cezara;
- d. stosowanie prostych metod szyfrowania przestawieniowego;

5. Stosowanie algorytmów przetwarzania ciągów i zbiorów danych, w szczególności:

- a. wyszukiwanie elementu w zbiorze;
- b. porządkowanie ciągu liczb, w tym metodami:
 - i. przez wstawianie,

- ii. bąbelkową,
- iii. przez wybieranie;
- c. wyznaczanie wartości elementów ciągów liczbowych metodami:
 - i. iteracyjnymi,
 - ii. rekurencyjnymi, w szczególności dla ciągu Fibonacciego;

6. Stosowanie wybranych technik algorytmicznych, w szczególności:

- a. stosowanie rekurencji, w szczególności analizowanie działania funkcji rekurencyjnych oraz wyznaczanie wartości funkcji dla zadanych argumentów;
- b. stosowanie podejścia zachłannego;
- c. analizowanie poprawności działania algorytmu;
- d. analizowanie działania algorytmu na podstawie jego implementacji;
- e. weryfikowanie poprawności algorytmu dla przykładowych danych;

7. Rozwiązywanie problemów algorytmicznych, w szczególności:

- a. rozwiązywanie problemów obliczeniowych;
- b. rozwiązywanie problemów optymalizacyjnych;
- c. rozwiązywanie problemów praktycznych, takich jak: wydawanie reszty najmniejszą liczbą nominatów.

II. PROGRAMOWANIE

Uczestnik powinien wykazywać się umiejętnością projektowania i tworzenia programów komputerowych w dopuszczonych językach programowania: C++; C#; Python. Zakres wymagań obejmuje umiejętności analizy problemów, projektowania algorytmów oraz implementowania rozwiązań w wybranym języku programowania.

1. Tworzenie programów rozwiązujących problemy algorytmiczne, w szczególności:

- a. implementowanie algorytmów;
- b. testowanie i uruchamianie programów;
- c. analizowanie poprawności działania programów;

2. Stosowanie podstawowych elementów języka programowania, w szczególności:

- a. instrukcji wejścia i wyjścia danych;
- b. zmiennych i typów danych;
- c. operatorów arytmetycznych;
- d. operatorów logicznych;
- e. operatorów porównania;

3. Stosowanie instrukcji sterujących, w szczególności:

- a. instrukcji warunkowych;
- b. instrukcji iteracyjnych (pętli);
- c. instrukcji sterujących przebiegiem programu;

4. Stosowanie podprogramów, w szczególności:

- a. definiowanie funkcji i procedur;
- b. stosowanie funkcji:
 - i. bez parametrów,
 - ii. z parametrami;
- c. zwracanie wartości z funkcji;

5. Rozwiązywanie problemów programistycznych, w szczególności:

- a. tworzenie programów rozwiązujących zadania obliczeniowe;
- b. tworzenie programów przetwarzających dane tekstowe;
- c. tworzenie programów przetwarzających ciągi liczbowe;

6. Stosowanie podstawowych funkcji przetwarzania struktur danych, w szczególności:

- a. tablic;
- b. napisów;
- c. list i struktur złożonych dostępnych w danym języku programowania.

III. ANALIZA I TESTOWANIE ROZWIĄZAŃ**1. Uczestnik powinien wykazywać się umiejętnością:**

- a. testowania programów dla różnych danych wejściowych;
- b. analizowania poprawności uzyskanych wyników;
- c. poprawiania błędów w programie.

Wykaz literatury i materiałów pomocniczych**Literatura**

1. Maciej M. Sysło, Algorytmy, Helion, 2016
2. Aditya Y. Bhargava, Algorytmy. Ilustrowany przewodnik, Helion, 2017
3. Eric Matthes, Python. Instrukcje dla programisty, Helion, 2019
4. Grzegorz Banaszak, Adam Dąbrowski, Programowanie w języku C++ dla szkół, Wydawnictwo Naukowe PWN (lub dowolny równoważny podręcznik języka C/C++)
5. Andrzej Daniluk, Programowanie w C#, Helion (lub dowolny równoważny podręcznik języka C#)

Materiały internetowe

1. Portal edukacyjny „Pasja Informatyki”;
<https://pasja-informatyki.pl/>
2. Portal „Algorytmy i programowanie”;
<http://www.algorytm.edu.pl/>
3. Platforma Edukacyjna Ministerstwa Edukacji Narodowej;
<https://zpe.gov.pl/>