Próbny Test 1

To jest próbny test, dla kandydatów zamierzających przystąpić do testu ECDL/ICDL z modułu Computing. Próbny test ma pokazać kandydatowi, jaka jest struktura i styl testu egzaminacyjnego, aby ułatwić rozwiązanie testu egzaminacyjnego.

W żadnym przypadku test próbny nie może być użyty w celach egzaminacyjnych

To jest próbny test ECDL / ICDL z modułu Computing. Próbny test zawiera 36 pytań. Za każdą poprawną odpowiedź można uzyskać 1 punkt. Maksymalna liczba punktów do zdobycia wynosi 36. Czas rozwiązania to 45 minut.

1 Myślenie komutacyjne to: **[1 pkt]**

a. Myślenie o komputerach i ile one kosztują. ☐

b. Analizowanie problemów i znajdowanie możliwych rozwiązań. ☐

c. Używanie kalkulatora do obliczania wartości wyrażeń matematycznych ☐

d. Sprawdzanie błędów w programie przed jego wersjonowaniem. ☐

2 Pojęcie **kod maszynowy** oznacza: **[1 pkt]**

a. Ciąg zer i jedynek utworzony z kodu źródłowego. ☐

b. Graficzny sposób reprezentacji algorytmu. ☐

c. Kilka prostych instrukcji, które należy wykonać jedna po drugiej. ☐

d. Opis objaśniający co program powinien robić i jak działać. ☐

3 Proces jasno określający problemy, które mają zostać rozwiązane podczas tworzenia nowego programu nazywamy: **[1 pkt]**

a. Projektowanie. ☐

b. Wprowadzanie poprawek. ☐

c. Programowanie. ☐

d. Analizowanie. ☐

4 Zostałeś poproszony o zaprojektowanie aplikacji online dla serwisu dostarczającego pizzę. Wskaż najważniejsze pytania, na które należy odpowiedzieć, aby rozwiązać ten problem. **[1 pkt]**

a. 1. Kto będzie dostarczał pizzę?
2. Jakie typy smartfonów będą wykorzystywane?
3. Czy system będzie zapamiętywał dane klientów? ☐
4. Jakie rodzaje pizzy i innych dodatków będą dostępne w menu?

b. 1. Ile będzie różnych sekcji z potrawami?
2. Czy system będzie zapamiętywał dane klientów?
3. Jakie rodzaje pizzy i innych dodatków będą dostępne w menu? ☐
4. Czy klienci będą mogli płacić online?

c. 1. Jakiego koloru będzie ekran?
2. Ile będzie różnych sekcji z potrawami?
3. Jakie rodzaje pizzy i innych dodatków będą dostępne w menu? ☐
4. Czy przepisy na różne typy pizzy będą w systemie?

d. 1. Ile przycisków powinna mieć aplikacja?
2. Jaki samochód będzie wykorzystany do przewozu pizzy? ☐
3. Czy klienci będą mogli płacić online?
4. Czy system będzie zapamiętywał dane klientów?

5 Zostałeś poproszony o napisanie systemu komputerowego do zarządzania rezerwacjami w hotelu. System będzie musiał rozwiązać takie problemy jak: przechowywanie informacji o gościach: numer pokoju i typ pokoju, a także zameldowanie i wymeldowanie gości. Który ze schematów można wielokrotnie wykorzystać w systemie. **[1 pkt]**

a. Procedurę wprowadzania nazwy hotelu na ekranie powitalnym systemu. ☐
b. Procedurę tworzenia konta gościa w systemie. ☐
c. Procedurę instalacji systemu na komputerze ☐
d. Procedurę odinstalowania systemu z komputera. ☐

6 W myśleniu komputacyjnym sekwencyjny zestaw instrukcji **[1 pkt]**

używanych do rozwiązywania problemów to:

a. Algorytm. ☐
b. Dekompozycja. ☐
c. Kod maszynowy. ☐
d. Procedura. ☐

7 Obrazowy sposób reprezentacji zestawu instrukcji rozwiązujących problem nazywamy? **[1 pkt]**

a. Funkcją. ☐
b. Kodem maszynowym. ☐
c. Schematem blokowym. ☐
d. Wyrażeniem logicznym. ☐

8 Specjalny typ zmiennej, która ma wpływ na działanie podprogramu to: **[1 pkt]**

a. Float. ☐
b. Funkcja. ☐
c. Parametr. ☐
d. Pseudokod. ☐

9 Fragment tekstu w kodzie, który wyjaśnia ludziom co kod robi nazywamy: **[1 pkt]**

a. Specyfikacją. ☐
b. Komentarzem. ☐
c. Tablicą. ☐
d. Rekurencją. ☐

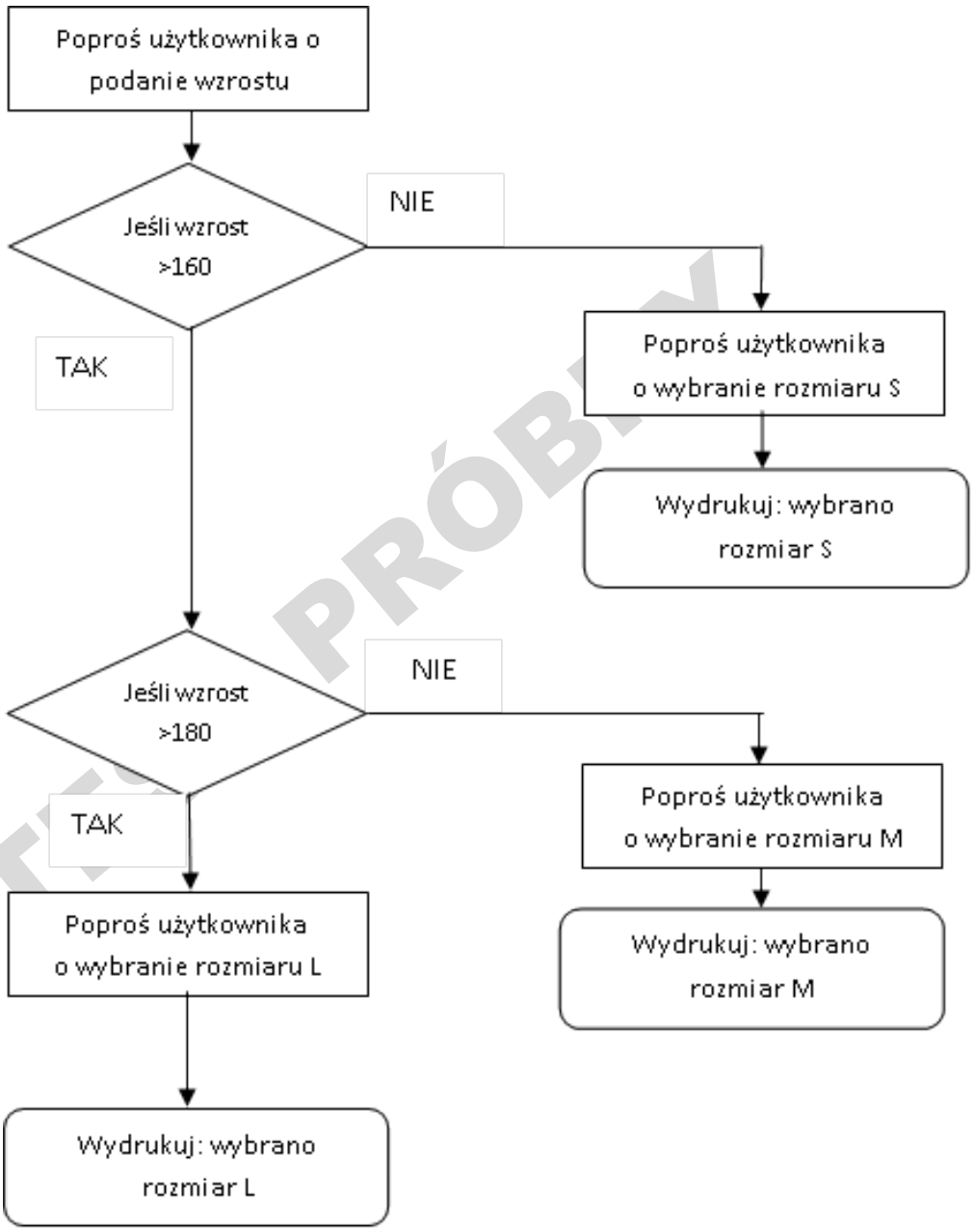
10 Który z poniższych elementów może być wykorzystywany w programie do przechowywania wartości, które mogą być wielokrotnie zmieniane? **[1 pkt]**

a. Komentarz. ☐
b. Zmienna. ☐
c. Pętla. ☐
d. Operator porównania. ☐

11 Która z poniższych nazw zmiennych powinna zostać wybrana, do przechowywania prywatnego kodu dostępu klienta? **[1 pkt]**

a. mojeImie. ☐
b. mojWiek. ☐
c. mojPIN. ☐
d. mojeDane. ☐

12 Poniższy schemat blokowy przedstawia: **[1 pkt]**



a. Instrukcje dotyczące projektowania trzech typów etykiet rozmiarowych. ☐

b. Instrukcje dla programu, który drukuje wysokość użytkownika w zależności od wybranego rozmiaru. ☐

c. Instrukcje dotyczące wprowadzania wzrostu do urządzenia do sprawdzania rozmiaru. ☐

d. Instrukcje dotyczące programu, który drukuje wymagany rozmiar, w zależności od wysokości użytkownika ☐

13 Wskaż operator logiczny dla wyrażenia „różne”. **[1 pkt]**

a. Not ☐
b. <= ☐
c. < ☐
d. != ☐

14 Fragment kodu powtarzany, gdy są spełnione pewne warunki nazywa się: **[1 pkt]**

a. Pseudokod. ☐
b. Komentarz. ☐
c. Wyrażenie logiczne. ☐
d. Pętla. ☐

15 Typ pętli używany w iteracjach to: **[1 pkt]**

a. Integer. ☐
b. While. ☐
c. Float. ☐
d. List. ☐

16 Które pojęcie określa podprogram dzielący problem na prostsze części i wywołujący samego siebie do rozwiązania tych prostszych części. **[1 pkt]**

a. Debugowanie. ☐
b. Rekurencja. ☐
c. Zdarzenie. ☐
d. Zmienna. ☐

17 Podaj nazwę instrukcji, która oblicza wartość wyrażenia logicznego, i na tej podstawie wybiera następne działanie. **[1 pkt]**

a. Warunkowa. ☐
b. Iteracjyjna. ☐
c. Złożona. ☐
d. Rekurencyjna. ☐

18 Podaj nazwę podprogramu, który wykonuje pewne akcje w programie bez zwracania wartości. **[1 pkt]**

a. Logiczny. ☐
b. Iteracyjny. ☐
c. Procedura. ☐
d. Zmienna. ☐

19 Podaj nazwę podprogramu, który oblicza wartość dla programu, który go zawiera.? **[1 pkt]**

a. Debuger. ☐
b. Procedura. ☐
c. Pętla for. ☐
d. Funkcja. ☐

20 Poniższy kod ma wypisać czas w formacie 17:22:56.

```
from time import strftime, gmtime

# Wpisz kod tutaj

print ( strftime() )
```

Wybierz linię kodu, która powinna zostać dodana, aby czas został wypisany w wymaganym formacie **[1 pkt]**

a. strftime("%H", gmtime()) ☐
b. strftime("%M", gmtime()) ☐
c. strftime("%Y", gmtime()) ☐
d. strftime("%X", gmtime()) ☐

21 Błąd polegający na niepoprawnym napisaniu konstrukcji w języku programowania nazywa się: **[1 mark]**

a. Składniowy. ☐
b. Rekurencyjny. ☐
c. Logiczny. ☐
d. Zmienna. ☐

22 Otwórz plik **Algorithm_Error.docx.** Algorytm w postaci schematu blokowego opisuje procedurę sprawdzania PINu użytkownika i blokowania systemu, gdy kod PIN zostanie podany błędnie trzy razy. Jednak w algorytmie został opuszczony jeden element. Zidentyfikuj błąd i wybierz poprawną wersję schematu blokowego. Zamknij plik **Algorithm_Error.docx**. **[1 pkt]**

a. Wersja A. ☐

b. Wersja B. ☐

c. Wersja C. ☐

d. Wersja D. ☐

23 Otwórz plik **Flowchart.docx**. Algorytm w postaci schematu blokowego częściowo opisuje procedurę **Regulamin kolejki górskiej**. Na podstawie dostarczonych informacji uzupełnij schemat blokowy algorytmu. Zapisz i zamknij plik **Flowchart.docx**. **[1 pkt]**

24 Otwórz plik **Division.py**. Zaktualizuj program, tak by **obliczał iloraz z dzielenia wartości rachunku przez liczbę osób**. Zapisz i zamknij plik **Division.py**. **[1 pkt]**

25 Otwórz plik **Comments.py**. Wstaw komentarz do programu powyżej linii **imie1 = "Janusz",** który poinformuje czytającego, że następne linie **kodu zdefiniują trzy imiona**. Zapisz i zamknij plik **Comments.py**. **[1 pkt]**

26 Otwórz plik **Initialising_String.py**. Wstaw kod do programu poniżej linii **# definicja i inicjacja zmiennej mojeZwierzatko** w którym zdefiniujesz zmienną o nazwie **mojeZwierzatko** i zainicjujesz ją używając imienia **Felix**. Zapisz i zamknij plik **Initialising_String.py**. **[1 pkt]**

27 Otwórz plik **Assign_Value.py**. Wstaw kod do programu poniżej linii **# przypisanie wartości do zmiennej cena,** który przypisze wartość **24** do zmiennej o nazwie **cena.** Zapisz i zamknij plik **Assign_Value.py**. **[1 pkt]**

28 Otwórz plik **Using_Integers.py**. Zmień kod programu tak, aby wartość **24** przypisana **zmiennej liczba** była **typu całkowitego**. Zapisz i zamknij plik **Using_Integers.py**. **[1 mark]**

29 Otwórz plik **Seasons.py**. Wstaw kod, który krotce (tupli) o nazwie **poryRoku** przypisze nazwy pór roku: **wiosna, lato, jesień, zima.** Zapisz i zamknij plik **Seasons.py**. **[1 pkt]**

30 Otwórz plik **Data_Output.py**. Program pyta użytkownika o jego wzrost. Zaktualizuj program tak, aby wypisywał wzrost użytkownika na ekranie. Zapisz i zamknij plik **Data_Output.py**. **[1 pkt]**

31 Otwórz plik **Boolean.py**. Zmodyfikuj program tak, aby wypisywał komunikat: **Liczba należy do przedziału**, jeżeli podana przez użytkownika liczba jest **mniejsza od 20, a większa lub rowna 13**. W przeciwnym wypadku powinien zostać wypisany komunikat**: Liczba nie należy do przedziału**.Użyj operatora **AND**. Zapisz i zamknij plik **Boolean.py**. **[1 pkt]**

32 Wstaw kod, który za pomocą instrukcji warunkowej **if ... else,** sprawdzi, czy uczeń zdał egzamin. Uczeń zdał egzamin, jeżeli **otrzymał wynik wyższy lub równy 75.** Zapisz i zamknij plik **If_Statement.py.** **[1 pkt]**

33 Otwórz plik **Function.py**. Zmodyfikuj program, aby zdefiniować funkcję o nazwie **Odejmowanie**, która obliczy różnicę dwóch liczb podanych przez użytkownika i zwróci wynik obliczeń. Zapisz i zamknij plik **Function.py**. **[1 pkt]**

34 Otwórz plik **Random.py**. Zmodyfikuj program tak, aby korzystając z biblioteki 'random' dostarczonej z Pythonem, wykorzystał dostępne funkcje do wygenerowania liczby losowej z przedziału **od 2 do 12 włącznie.** Zapisz i zamknij plik **Random.py**. **[1 pkt]**

35 Otwórz plik **Syntax.py**. Znajdź w programie i popraw jeden błąd w pisowni i jeden błąd interpunkcyjny. Zapisz i zamknij plik **Syntax.py**. **[1 pkt]**

36 Otwórz plik **Logic.py** . Znajdź w programie i popraw jeden błąd logiczny i jeden błąd typu danych. Zapisz i zamknij plik **Logic.py**.

Zapisz i zamknij wszystkie otwarte pliki oraz zamknij wszystkie otwarte aplikacje. **[1 pkt]**

To już koniec testu
Jeżeli masz jeszcze czas sprawdź poprawność odpowiedzi.