

## Spis treści

<b>I. Programowanie niestukturalne</b>	<b>3</b>
1. Instrukcje break i continue	4
2. Instrukcja goto	7
3. Instrukcja switch	8
4. Optymalizacja	13
<b>II. Preprocesor</b>	<b>16</b>
1. Dyrektywa preprocesora	16
1.1. Dyrektywa #pragma	16
1.2. Dyrektywa include	16
1.3. Dyrektywy #if, #ifdef, #ifndef, #else, #endif	18
1.4. Dyrektywy #define i #undef	20
2. Zastosowania dyrektywy #define	23
2.1. Definiowanie „stałych”	23
2.2. Makrodefinicje	25
2.3. Inne zastosowania definicji identyfikatorów	27
<b>III. Formatowanie wejście/wyjście</b>	<b>33</b>
1. Funkcje ze zmienną liczbą argumentów	32
2. Funkcje formatowanego wyjścia	37
2.1. Funkcja printf	37
2.2. Funkcja fprintf	40
2.3. Funkcja sprintf	41
2.4. Funkcje vprintf, vfprintf, vsprintf	41
2.5. Funkcja cprintf	43
3. Funkcje formatowanego wejścia	44
3.1. Funkcja scanf	44
3.2. Standardowe wejście	49
3.3. Funkcja fscanf	50
3.4. Funkcja sscanf	54
3.5. Funkcja cscanf	56
3.6. Funkcje vscanf, vfscanf i vsscanf	56
4. Zastosowania funkcji formatowanego wejścia/wyjścia	57
4.1. Projektowanie wyglądu ekranu	57
4.2. Reprezentacja rekordów w pliku	59
<b>IV. Programowanie współbieżne</b>	<b>62</b>
1. Dlaczego współbieżność?	62
2. Funkcje setjmp i longjmp	63
3. Przełączanie zadań	64
4. Zapis praktyczny	66
5. Program współbieżny	70
6. Komunikacja między procesami	75
7. Współbieżne wejście z klawiatury	80
<b>V. Kod wynikowy</b>	<b>83</b>
1. Zmieniamy Startup	84
2. Programy rezydentne	88
2.1. Jakie są zastosowania programów rezydentnych?	88
2.2. Tworzenie programów rezydentnych	89
2.3. Funkcja main jako program rezydentny	89
3. Przykładowe programy rezydentne	94