

Практика 1-го курсу

1. Результатом практики є завершена програма для розв'язання визначеної задачі та звіт по програмі.
2. Програма має бути написана на одній з мов програмування C, C+, Visual C, Pascal, Delphi з можливим використанням об'єктів OLE Automation, COM, Flash, ActiveX тощо.
3. Звіт подається в роздрукованому та скріпленому вигляді та містить такі елементи:
 - a. вступ (загальна характеристика роботи);
 - b. постановка задачі (мета та розв'язання задачі);
 - c. алгоритми програми з описом основних функцій та процедур;
 - d. опис роботи програми;
 - e. можливі помилки при використанні програми;
 - f. використана література.
4. Програма подається на дискеті, CD-R або CD-RW диску. На носії інформації має бути створена папка, яка має назву прізвища виконавця. В ній має знаходитись:
 - a. скопійований файл запуску програми з назвою Start.exe;
 - b. файл Readme.txt, що містить інформацію по запуску програми;
 - c. допоміжні файли (при необхідності);
 - d. папка Source, де знаходиться програма з відкритим кодом;
 - e. папка Звіт, що містить файл – електронну версію звіту.
5. Всі носії інформації мають бути чітко і розбірливо підписані.
6. При написанні програми слід врахувати такі основні фактори:
 - a. правильність роботи;
 - b. дизайн програми;
 - c. ефективність алгоритму;
 - d. простота роботи користувача;
 - e. наявність довідкової системи;
 - f. інформаційне супроводження програми.
7. Захист практики відбудеться в середині вересня 2005 року. Попередній захист проводитиме керівник практики з 5 по 9 вересня, тому всі роботи повинні бути здані до 2 вересня 2005 року. **Запізнення зі здачею документів є підставою для зниження оцінки!**
8. Студент має право самостійно вибрати задачу для розв'язання (за умови погодження з науковим керівником). Зареєструвати вибір задачі необхідно у наукового керівника практики до 30 червня 2005 року. Відсутність реєстрації до вказаної дати є підставою для недопущення студента до захисту практики!
9. Всі програми, звіти та допоміжні матеріали повинні бути красиво і акуратно оформлені, виконані на українській мові. При проблемах з кодуванням, як виключення, дозволяється використовувати в програмах англійську мову (програми, написані на „трансліті” **не приймаються**).
10. При включенні до програми додаткових нестандартних Windows-об'єктів необхідно на носії інформації розмістити необхідні інсталяційні пакети.

Приблизні теми:

1. Розв'язати довільну систему лінійних рівнянь методом Гауса.
2. Дано файл з текстом. Написати програму, що розбиває текст на закінчені фрагменти тексту, кожен з яких не перевищує наперед заданої кількості символів, та дає смислову назву кожному з них. Передбачити різні стилі тексту (заголовки, списки, таблиці тощо).
3. Змоделювати систему деякого міста. До програми вводяться 2D рисунки будівель та інших об'єктів (дороги, річки, мости тощо) різного заданого розміру, а програма має по них будувати 3D карту міста.
4. Написати програму для розв'язання японських кросвордів.
5. Написати програму, що вилучає з усіх файлів (розширення *.html або *.htm) даної директорії вказані користувачем теги.
6. Написати програму, що вилучає з усіх файлів (розширення *.html або *.htm) даної директорії вказані користувачем елементи (наприклад, повтори, зайву інформацію тощо).
7. Написати програму, що створює HTML документ з меню довільного рівня складності (горизонтальне, вертикальне, ієрархічне, на декількох шарах тощо).
8. Написати програму для знаходження похідної та первісної довільної функції.
9. Дано N міст з географічними координатами їх розташування. Провести найменшу кількість доріг (по довжині), щоб з будь-якого міста можна було дійхати до будь-якого іншого.
10. Написати програму-гру у доміно для 4 гравців.
11. Знайти (принаймні наближено) всі розв'язки довільного поліноміального рівняння.
12. Знайти оптимальний хід в грі з двох учасників, де позиція в будь-який момент може бути оцінена наперед заданою функцією.
13. В вершинах великого (не менше 3 млн. вершин) графу розташовані числа. Правильною оцінкою вершини називається найбільше число парного рівня та найменше число непарного рівня для всіх вершин, що розташовані нижче. Знайти правильну оцінку (наближену) початкової вершини графу за обмежений (заданий заздалегідь) час.
14. На гору піднімаються N альпіністів, кожен з яких може нести вагу w , а споживає кожного дня a продуктів. Визначити оптимальний план сходження на вершину та спуску з неї, якщо альпіністи мають право передавати продукти.
15. Для текстового файлу передбачити операції пошуку даних, їх аналітичної заміни за найменший час.
16. Магазин надає значні знижки при купівлі товарів оптом. Розробити програму, яка попереджає про всі неоптимальні для покупця дії.
17. Написати програму, що здійснює обхід довільної дошки $n \times n$ шаховим конем.
18. Написати програму, що виконує будь-які дії зі звичайними дробами (+, -, *, /, ^, тощо).
19. Написати програму, що виконує арифметичні операції з римськими цифрами.
20. Цифри від 0 до 9 розташовані у деякому визначеному порядку. Розставити між ними арифметичні знаки таким чином, щоб була отримана заданий результат.

21. Змодельовати екосистему острова (розміру $N \times N$), на якому живуть вовки та кролики. Кожен кролик перетворюється на 2 кроликів з ймовірністю P . Вовк має опинитися в одному квадраті з вовчицею для народження нового вовка чи вовчиці. Якщо вовк не з'їв протягом N ходів кролика, він вмирає.
22. Розставити на шаховій дошці найбільшу кількість фігур таким чином, щоб фігури одного рівня не загрожували один одному.
23. По заданих K числах реалізувати побудову магічного прямокутника $M \times N$.
24. Написати програму, що розв'язує довільний числовий ребус.
25. Нехай масив $S(m)$ - відомість видачі зарплати за один місяць. Касир видав суму купюрами різних номіналів, інформація про які зберігається у масиві $L(n)$, а у масиві $K(n)$ – кількість знаків кожного достоїнства ($\sum_{i=1}^m s_i \leq \sum_{j=1}^n k_j l_j$). Чи зможе касир роздати кожному зарплату без додаткового розміну грошей?
26. Є n деталей, кожна з яких проходить опрацювання спочатку на однім верстаті, потім на іншому. На кожному верстаті одночасно обробється тільки одна деталь; часу на переналагодження не потрібно. Відомий час опрацювання деталі на кожному верстаті. Упорядкувати деталі так, щоб сумарний час опрацювання партії деталей був мінімальним.
27. Написати програму для визначення кількості $2 \times N$ -значних „щасливих” квитків. „Щасливий” квиток має визначатися за декількома способами.