|  |  |
| --- | --- |
| ***Варіант 1***  1)Директива #define має настпуну структуру:  #define ідентифікатор\_макро тіло\_макро  Препроцес переглядає текст С-програми і замінює кожне входження ідентифікатора\_макро на тіло макророзширення. Заміні не підлягають ідентифікатору\_макро розміщені у літералах  2) Алгоритм сортування вибором працює таким чином:  Знаходить у списку найменше значення  міняє його місцями із першим значенням в списку  Повторює два попередніх кроки, доки список не завершиться(починаючи з другої позиції)  64 25 12 22 11  11 25 12 22 64  11 12 25 22 64  11 12 22 25 64  5) ofstream file("oldmast.dat",ios\_base::out); ofstream \*ofPtr = &file;  6)fscanf(fPtr,”%d%s%d\n”,&account,company,&amount);  7)Ні, не може  8)fread()  9)Г)ПОмилка компіляції — не ініціалізована змінна sss  10)Помилка компіляції  11)Ні  12)- | Варіант 2  ***1)***Статичний тип пам'яті  Статичні об'єкти створюються компонувальником і живуть весь час виконання програми. Статичні змінні (клас static) поділяються на змінні з трьома типами зв'язування: зовнішнє зв'язування, внутрішнє зв'язування та без зв'язування. Незалежно від типу зв'язування такі змінні працюють на протяці всього часу викоання програми і компілятор резервує фіксований блок пам'яті для збереження всіх статичних змінних  ***2) Обчислювальна складність алгоритму –*** це приблизна кількість значимихоперацій, які виконуються алгоритмом. Обчислювальна складність характеризує (вимірює) відносний час виконання алгоритму. Реальний час, необхідний для розв’язку задачі, непридатний для аналізу ефективності алгоритму, оскільки алгоритм не стає кращим, якщо його перенести на більш  швидкий комп'ютер, або гіршим, якщо його виконувати на більш повільному комп’ютері. Насправді, фактична кількість операцій алгоритму на тих чи інших вхідних даних не представляє великого інтересу і не дає нам багато інформації про алгоритм. При аналізі алгоритмів нас буде цікавити швидкість росту алгоритму.  5)char name[100];  int AccountSum;  char name[100];  float currentBalance;  FILE \*fp;  fp=fopen(“oldmast.dat”);  fscanf(fp”%d%s%d\n”,&AccountSum,name,&currentBalance);  6) має бути а+  7) Правильно,оскільки при прямому доступі можна виконувати операції ввід,вивід використовуючи сисстему ввід,вивід мови С та функцію fseek(),яка встановлює вказівник поточної позиції в файлі.  8) fgets();  9)Г)  10)Б)  11)Помилка етапу компіляції  12)1 |
| ***Варіант 3***  1) auto. Этот спецификатор автоматического класса памяти указывает на то, что объект располагается в локальной памяти. Он используется в операторах объявления в теле функций, а также внутри блоков операторов. Объекты, имена которых объявляются со спецификатором auto, размещаются в локальной памяти непосредственно перед началом выполнения функции или блока операторов. При выходе из блока или при возвращении из функции , соответствующая область локальной памяти освобождается и все ранее размещённые в ней объекты уничтожаются.  1)Будь-яка функція в програмі мовою С може бути викликана рекурсивно,тобто вона може викликати саму себе. Функція називається рекурсивною, якщо під час її обробки виникає її повторний виклик, або безпосередньо, або побічно, шляхом ланцюжка викликів інших функцій. Прямою (безпосередньою) рекурсією є виклик функції усередині тіла цієї функції.Непрямою рекурсією є рекурсія, що здійснює рекурсивний виклик функції за допомогою ланцюжка виклику інших функцій.  2)struct Single\_List {  int Data;  Single\_List \*Next;};  Single\_List \*Head;  void Make\_Single\_List(int n,Single\_List\*\* Head){  if (n > 0) {  (\*Head) = new Single\_List();  //выделяем память под новый элемент  cout << "Введите значение ";  cin >> (\*Head)->Data;  //вводим значение информационного поля  (\*Head)->Next=NULL;//обнуление адресного поля  Make\_Single\_List(n-1,&((\*Head)->Next));  }//создание циклического однонаправленного списка  void Make\_Circle\_Single\_List(int n,  Circle\_Single\_List\*\* Head,Circle\_Single\_List\* Loop){ | if (n > 0) {ж  (\*Head) = new Circle\_Single\_List();  //выделяем память под новый элемент  if (Loop == NULL) Loop = (\*Head);  cout << "Введите значение ";  cin >> (\*Head)->Data;  //вводим значение информационного поля  (\*Head)->Next=NULL;//обнуление адресного поля  Make\_Circle\_Single\_List(n-1,&((\*Head)->Next),Loop);  }  else {  (\*Head) = Loop;  }}}  5)ofstream file("newmast.dat",ios::out||ios::trunc);  ofstream \*nfPtr = &file;  6)Правильне твердження  функція fscanf використовується для читання даних з файлу, а не зі стандартного вводу  7)Хибне твердження  int fseek(FILE \*stream, long int offset, int whence);  fseek зміщує вказівник в файлі stream на offset разів відносно аргумента whence/  8)fgets();  9)Помилка компіляції  10)і = -128  11)Зациклення  12)В мейні значенню ігрик не було нічого присвоєно |
| ***Варіант 4***  1)struct Double\_List {  int Data;  Double\_List \*Next, \*Prior;  }; Double\_List \*Head;  void Make\_Double\_List(int n,Double\_List\*\* Head,  Double\_List\* Prior){  if (n > 0) {  (\*Head) = new Double\_List();  cout << "Введите значение ";  cin >> (\*Head)->Data;  (\*Head)->Prior = Prior;  (\*Head)->Next=NULL;  Make\_Double\_List(n-1,&((\*Head)->Next),(\*Head));  }  else (\*Head) = NULL;  }  //создание циклического двунаправленного списка  Circle\_Double\_List\* Make\_Circle\_Double\_List(int n,  Circle\_Double\_List\*\* Head,Circle\_Double\_List\* Loop){  Circle\_Double\_List\* ptr;  if (n > 0) {  (\*Head) = new Circle\_Double\_List();  if (Loop == NULL) Loop = (\*Head);  cout << "Введите значение ";  cin >> (\*Head)->Data;  (\*Head)->Next=NULL;  ptr = Make\_Circle\_Double\_List(n-1,&((\*Head)->Next),Loop);  if ((\*Head)->Next != NULL)  (\*Head)->Next->Prior = (\*Head);  if ((\*Head)->Prior == NULL)  (\*Head)->Prior = ptr;  if (ptr == NULL)  return \*Head; | else return ptr;  }  else {  (\*Head) = Loop;  return NULL;  }  }  2)функція без параметрів getchar() вводить символ і повертає ціле число – код введеного символа. Для виводу символу використовується функція putchar(int). Функція gets (char\*), з параметром – вказівником на стрічку, на відміну від scanf() зчитує всі символи з буфера вводу, включаючи символ ENTER (‘\n’), заміняючи останній на символ ‘\0’. Ця властивість scanf() часто використовується при виділенні слів у тексті та підрахунку числа слів у стрічці. Функція puts (char\*) виводить на екран всю стрічку, на початок якої вказує char\*, включно із символом ‘\0’, замінивши останній на (‘\n’)  5)ofstream f; f.write((char\*)&accountSum, sizeof(accountSum));  f.write(name,strlen(name));  f.write((char\*)&currentBalance,sizeof(currentBalance));  6)Так оскільки,це дуже зручно,ми можемо перевіряти чи файл відкрився вдало. ***if* ( (*fp=fopen*(“*D:\\lib1\\res1.dat*“,”*rb*+“) != *NULL*)**  7)Ні, записи у файлі прямого доступу мають однакову довжину  8)fopen( “шлях до файлу”б метод доступу до даних файлу)  9) і = -128 10) а)0 11) правильно 12) 100100101 |
| ***Варіант 5***  1)width(int x)-мінімальне число знаків до наступного виводу  fill(char x)-встановлює символ-заповнювач і повертає попередній символ-заповнювач. За умовчанням як символ-заповнювач використовується пропуск  precision(int x)-встановлює число значущих знаків для чисел з плаваючою крапкою.  endl-перевод рядка  flush-вивантажує вміст буфера в потік  dec-міняє формат виведення числа на десятковий  oct-міняє формат виведення числа на вісімковий  hex-міняє формат виведення числа на шістнадцядковий  ws-витягує і видаляє символи порожніх проміжків з потоку  showpos-показує '+' перед додатними числами  showpoint-показує десяткову крапку  noshowpoint-приховує десяткову крапку  2)Сортування методом злиття.Масив ділиться навпіл, до кожної половини застосовується рекурсивно та сама процедура сортування злиттям, а відсортовані частини з'єднуються в один впорядкований масив. Сортування методом злиття потребує виконання О(n log n) ,базових операцій над елементами масиву розмірності n.  5) File\* fPtr;  fPtr=fopen(“111.dat”);  fprintf(fPtr,”%d%s%d\n ”,account,company,amount);  6)Операція закриття файла не є обовязковою – при завершенні роботи програми файл закривається аварійно, проте, на відміну відфункції *fclose* (), не проводиться розвантаження буферів вводу-виводу, щоможе привести до втрати частини даних.  7)Масив  8)Функція fseek (вказівник на файл, величина зміщення, точка відліку для зміщення) перестановлює вказівник файла в задане положення.  9) str\_1 == c;10)6 11)-128 12)Помилка комп.(межі масиву) | ***Варіант 6***  1)  fopen () ------------ feof ()  fclose () ------------fprintf ()  fputc () ------------ fscanf ()  fgetc ()------------ fputs ()  fgets () ------------ fseek ()  2)**Послідовний пошук** — послідовний перегляд всіх елементів списку, поки не знайдеться потрібний  **Бінарний пошук —** ключ V порівнюється із середнім елементом списку і якщо ці значення рівні, то елемент знайдений, якщо ні, то продовжується пошук в одній з половин списку  **М-блоковий пошук — цей** спосіб зручний при індексному зберенні списку. Передбачається, що початковий упорядкований список В довжини Н розбитий на М підсписків В1, В2, ...Вм з довжинами Н1, ... Нм відповідно таким чином, що В = В1,...,Вм.  Для ключа V потрібно спочатку визначити перший із списків Ві останній елемент якогобільший V, а потім застосувати послідовний пошук до списку Ві  5) fopen();  6) Потрібно сокристуватися ф-цією fseek();  7) fclose();  8) У стеці елементи додаються і видаляються тільки з вершини стеку, а в чернах — елементи видаляються з початку списку, а додаються в кінці  9)a  10) б  11) б  12) мама |