|  |
| --- |
| ОП 2020  **Вариант** - 1 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **void StudSortOklad(const char \* FileName )**  **{**  **Student \* pStudMas;**  **int Razm; // число записей**  **// Чтение из файла в массив**  **StudFileToMas( FileName , &pStudMas , &Razm );**  **// Сортировка массива по окладу по убыванию**  **for (int i = 0 ; i < Razm - 1 ; i++ )**  **for (int k = 0 ; k < Razm - 1 ; k++ )**  **if ( ((Student \*) (pStudMas + k))->Oklad >**  **((Student \*) (pStudMas + k +1))->Oklad )**  **{ SwapStudent ( pStudMas + k , pStudMas + k+1 ); };**  **// Запись в файл**  **StudMasToFile( FileName , pStudMas , Razm );**  **};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 2 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **void StudSortName(const char \* FileName )**  **{**  **Student \* pStudMas;**  **int Razm; // Число записей**  **// Чтение из файла в массив**  **StudFileToMas( FileName , &pStudMas , &Razm );**  **for (int i = 0 ; i < Razm - 1 ; i++ )**  **for (int k = 0 ; k < Razm - 1 ; k++ )**  **if ( strcmp(((Student \*) (pStudMas + k))->Name ,**  **((Student \*) (pStudMas + k +1))->Name) > 0 )**  **{ SwapStudent ( pStudMas + k , pStudMas + k+1 ); };**  **// Запись в файл**  **StudMasToFile( FileName , pStudMas , Razm );**  **};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 3 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **void Summ6 (int \* iMas, long \* sum)**  **{**  **int i = 0;**  **while (iMas[i] != 0)**  **{**  **\*sum = \*sum + iMas[i];**  **i++;**  **}**  **}** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 4 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **// Описание массивов и их инициализация**  **int B[5][3] ={ { 1,2,3}, { 1,2,3},{ 1,2,3}, { 1,2,3},{ 1,2,3} };**  **int C[3][6] ={ { 1,2,3,4,5,6}, { 1,2,3,4,5,6},{ 1,2,3,4,5,6} };**  **int D[5][6];**  **//**  **for (int i = 0 ; i <5 ; i++ )**  **for (int k = 0 ; k <6 ; k++)**  **{**  **D[i][k] = 0;**  **for (int j = 0 ; j <3 ; j++)**  **D[i][k] = D[i][k] + B[i][j]\*C[j][k] ;**  **};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 5 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **// Исходный массив**  **int iMas[] = {3,3,2,4,5,6,0,1,9};**  **int Razm = sizeof(iMas)/sizeof(int);**  **int Flag = 0;**  **for (int k= 0 ; k<Razm - 1 ;k++ )**  **{**  **Flag = 0;**  **for (int i = 0 ; i <Razm - k - 1 ; i++)**  **{**  **if ( iMas[ i ] > iMas[ i+1] ) //**  **{ Swap( &iMas[ i ], &iMas[ i+1] ); Flag = 1;}**  **};**  **if (Flag == 0) break;**  **};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 6 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **// Исходный массив**  **int iMas[] = {3,3,2,4,5,6,0,1,9};**  **int Razm = sizeof(iMas)/sizeof(int);**  **int MinMax = iMas[0];**  **for (int i = 0 ; i <Razm ; i++ )**  **{**  **MinMax = iMas1[i];**  **Num = i;**  **Flag = 0;**  **for (int k= i ; k<Razm ;k++ )**  **if ( MinMax > iMas1[k] ) { Num = k; MinMax = iMas1[k];**  **Flag = 1; };**  **if ( Flag == 1)**  **Swap( &iMas1[ i ], &iMas1[ Num] );**  **};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 7 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **char Words[20][40]; // Двумерный массив символов**  **char TempWord[20] = "";**  **char Buf[800]; // Строка для поиска слов**  **int l = 0;**  **int k = 0;**  **for (int i = 0 ; (unsigned)i <= strlen(Buf); i++ )**  **{**  **TempWord[l] = Buf[i];**  **if ( Buf[i] != ' ' )**  **{ l++;}**  **else**  **{ TempWord[l] = '\0';**  **strcpy\_s(Words[k], 20, TempWord ); k++; l = 0;}**  **};**  **if (l != 0) {TempWord[l] = '\0';**  **strcpy\_s(Words[k], 20, TempWord );};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 8 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **void FindStudNum( const char \* FileName , Student \* pS , int Numb){**  **Student \* pStudMas;**  **int Razm;**  **StudFileToMas( FileName , &pStudMas , &Razm );**  **if ( Numb < 0 || Numb >= Razm )**  **{**  **pS->Name[0] = '\0';**  **pS->Num = NULL;**  **pS->Oklad = NULL;**  **}**  **else**  **memcpy( pS , pStudMas + Numb , sizeof(Student) );**  **delete [] pStudMas;**  **};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 9 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **char \* SubString( char \* string,int start , int length)**  **{**  **char Temp[30];**  **int Razm = strlen ( string );**  **if ( ( start + length +1) >= Razm ) {**  **strcpy(Temp, "ERROR: Ошибка длины!");**  **Temp[ 20 ] = '\0'; ;return Temp; }**  **if ( start < 0 || length < 0 ) {**  **strcpy(Temp, "ERROR: Ошибка параметров!");**  **Temp[ 25 ] = '\0'; ;return Temp; }**  **else strncpy\_s (Temp, Razm , string+ start , length);**  **Temp[ length ] = '\0';**  **return Temp;**  **};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 10 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **#define RazmMas 5**  **char StrMas[RazmMas][10]={"Сидоров", "Алетров", "Иванов", "Жучков" , "Акулов"};**  **//**  **for (int k = 0 ; k < RazmMas - 1 ; k++)**  **for ( int i =0 ; i < RazmMas - 1; i++ )**  **{**  **if ( strcmp(&StrMas[i][0] , &StrMas[i +1][0]) > 0 )**  **SwapStr(&StrMas[i][0] , &StrMas[i +1][0]);**  **};**  **//**  **for (int i =0 ; i < RazmMas ; i++ )**  **printf (" %d. - %s \n" , i + 1 , &StrMas[i][0]);** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 11 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **struct Student {**  **char \* pName;**  **char \* pFam;**  **int Kurs;**  **float Stipen; };**  **//**  **void ChangeStudent( Student \* S,const char \* pFam ,const char \* pName, int Kurs , float Stipen)**  **{**  **if ( Kurs != 0) S->Kurs = Kurs;**  **if ( Stipen != 0.0f ) S->Stipen = Stipen;**  **if ( pFam != NULL )**  **{**  **if (S->pFam != NULL) delete S->pFam;**  **S->pFam = new char [strlen (pFam) + 1];**  **strcpy( S->pFam , pFam);**  **};**  **if ( pName != NULL )**  **{**  **if (S->pName != NULL) delete S->pName;**  **S->pName = new char [strlen (pName) + 1];**  **strcpy( S->pName , pName);**  **};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 12 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **struct Student {**  **char \* pName;**  **char \* pFam;**  **int Kurs;**  **float Stipen; };**  **//**  **void InitStudent(Student \* pS,const char \* pFam ,const char \* pName, int Kurs , float Stipen){**  **pS->Kurs = Kurs;**  **pS->Stipen = Stipen;**  **if ( pFam != NULL )**  **{ pS->pFam = new char [strlen (pFam) + 1];**  **strcpy( pS->pFam , pFam);**  **};**  **if ( pName != NULL )**  **{ pS->pName = new char [strlen (pName) + 1];**  **strcpy( pS->pName , pName);**  **};};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 13 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **struct Student {**  **char \* pName;**  **char \* pFam;**  **int Kurs;**  **float Stipen; };**  **//**  **for( int k =0 ; k< Rzm - 1 ; k++)**  **{**  **for ( int i =0 ; i< Rzm - 1 ; i++ ) {**  **if ( strcmp((pPotok + i)->pFam , (pPotok + i + 1)->pFam ) > 0 ) //**  **{ SwapStudent( &pPotok[i] , &pPotok[i + 1] );**  **Flag =1; };**  **};**  **if ( Flag == 0 ) break; };** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 14 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **struct ListElem{**  **ListElem \* pNext;**  **int ListVal; };**  **// Структура простого списка**  **struct List {**  **ListElem Head; // Голова списка**  **ListElem Tail; // Хвост списка**  **int Count; }; // Счетчик элементов**  **//**  **void AddList( List \* pL , ListElem \* pE)**  **{**  **if (pL->Head.pNext == NULL) {pL->Head.pNext = pE;**  **pL->Tail.pNext = pE;}**  **else**  **{**  **pE->pNext = pL->Head.pNext;**  **pL->Head.pNext = pE; };**  **(pL->Count)++; };** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 15 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **struct ListElem{**  **ListElem \* pNext;**  **int ListVal; };**  **// Структура простого списка**  **struct List {**  **ListElem Head; // Голова списка**  **ListElem Tail; // Хвост списка**  **int Count; }; // Счетчик элементов**  **//**  **void DelList( List \* pL )**  **{ if (pL->Head.pNext == NULL) { return; }**  **else**  **{ ListElem \* pE;**  **pE = pL->Head.pNext ;**  **pL->Head.pNext = pL->Head.pNext->pNext;**  **pE->pNext = NULL;**  **if (pL->Head.pNext == NULL) { pL->Tail.pNext = NULL; };**  **(pL->Count)--; };**  **};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 16 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **struct ListElem{**  **ListElem \* pNext;**  **int ListVal; };**  **// Структура простого списка**  **struct List {**  **ListElem Head; // Голова списка**  **ListElem Tail; // Хвост списка**  **int Count; }; // Счетчик элементов**  **//**  **void PrintElemList ( ListElem E )**  **{**  **printf ("Печать списка элементов: \n");**  **ListElem \* pE = &E ;**  **while ( pE != 0)**  **{**  **printf ("Элемент = %d \n", pE->ListVal );**  **pE = pE->pNext; // Очень важно - навигация по списку**  **};**  **};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 17 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **…**  **srand( (unsigned)time( NULL ) );**  **rand();**  **for ( int i =0 ; i< 10 ; i++ )**  **{**  **int n = (rand()\*99)/ RAND\_MAX + 1; // 1 - 100**  **printf("Случайное число %2d. %d\n" , i+1 , n );**  **};**  **…** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 18 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **void StudFileToMas( const char \* FileName , Student \*\* pMas , int \* pRazm)**  **{ long posF;**  **int pFBin = \_open( FileName, \_S\_IREAD |\_O\_BINARY );**  **if ( pFBin == -1 ) { \*pRazm = NULL; return;};**  **posF = \_lseek( pFBin, 0, SEEK\_END ); //**  **\*pRazm = posF / sizeof(Student);**  **\*pMas = new Student[ \*pRazm];**  **posF = \_lseek( pFBin, 0, SEEK\_SET ); // На начало**  **// Буфер для чтения записи**  **Student SBuf;**  **for (int i = 0 ; i < \*pRazm ; i++ )**  **{**  **int nByte = \_read( pFBin , &SBuf , sizeof(Student));**  **if ( nByte == NULL) break;**  **memcpy( \*pMas + i , &SBuf , sizeof(Student)); };**  **\_close( pFBin );**  **};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 19 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **void StudMasToFile( const char \* FileName , Student \* pMas , int Razm)**  **{**  **char Comand[40];**  **// Снятие защиты с файла - отключение Readonly**  **strcpy (Comand , "attrib -R ");**  **strcat (Comand , FileName);**  **system( Comand );**  **int pFBin = \_open( FileName , \_O\_RDWR |\_O\_BINARY | \_O\_CREAT | \_O\_TRUNC);**  **if ( pFBin == -1 ) return;**  **// цикл записи**  **for ( int i = 0 ; i < Razm ; i++ )**  **{**  **\_write (pFBin , pMas + i , sizeof(Student));**  **};**  **\_close( pFBin );**  **};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 20 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **double StudSumm( const char \* FileName , double \* Sum)**  **{**  **Student \* pStudMas;**  **int Razm;**  **\*Sum = 0.0;**  **StudFileToMas( FileName , &pStudMas , &Razm );**  **for ( int i = 0 ; i < Razm ; i++)**  **\*Sum+= (pStudMas + i)->Oklad;**  **delete [] pStudMas;**  **return \*Sum;**  **};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 21 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **void ChangeStudNum( const char \* FileName , Student pS , int Numb){**  **Student \* pStudMas;**  **int Razm;**  **StudFileToMas( FileName , &pStudMas , &Razm );**  **if ( Numb < 0 || Numb >= Razm )**  **{**  **delete [] pStudMas;**  **return;**  **}**  **else**  **memcpy( pStudMas + Numb , &pS, sizeof(Student) );**  **StudMasToFile( FileName , pStudMas , Razm );**  **delete [] pStudMas;**  **};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 23 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **void SwapStudFile( const char \* FileName , int NumA ,int NumB)**  **{**  **Student \* pStudMas;**  **int Razm;**  **int i ;**  **StudFileToMas( FileName , &pStudMas , &Razm );**  **if ( NumA < 0 || NumA >= Razm || NumB < 0 || NumB >= Razm )**  **{ }**  **else**  **{SwapStudent ( pStudMas + NumA , pStudMas + NumB );**  **StudMasToFile( FileName , pStudMas , Razm );};**  **//**  **delete [] pStudMas;**  **};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 20 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **void StudDelNum(const char \* FileName , int Numb )**  **{**  **Student \* pStudMas;**  **int Razm;**  **StudFileToMas( FileName , &pStudMas , &Razm );**  **if ( Numb < 0 || Numb > Razm ) return;**  **if ( Numb == 0) {**  **StudDel( FileName , First );**  **delete [] pStudMas;**  **return;};**  **if ( Numb == Razm -1) {**  **StudDel( FileName , Last );**  **delete [] pStudMas;**  **return;**  **};**  **????** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 16 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **struct ListElem{**  **ListElem \* pNext;**  **int ListVal;**  **};**  **// Структура простого списка**  **struct List {ListElem Head; // Голова списка**  **ListElem Tail; // Хвост списка**  **int Count; // Счетчик элементов**  **};**  **void AddTailList( List \* pL , ListElem \* pE)**  **{ if (pL->Head.pNext == NULL) {pL->Head.pNext = pE;**  **pL->Tail.pNext = pE;}**  **else**  **{ ListElem \* pELast;**  **ListElem \* pCurr;**  **pCurr = &(pL->Head) ;**  **while ( pCurr->pNext != NULL )**  **{ pELast = pCurr;**  **pCurr = pCurr->pNext; };**  **pELast->pNext = pCurr;**  **pCurr->pNext = pE;**  **pL->Tail.pNext = pE;**  **pE->pNext = NULL;**  **};**  **(pL->Count)++ ;**  **};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - |
|  |
|  |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 1 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу :**  **int Summ5 (int \* mas ,int k ,int \* psum)**  **{**  **// НУ цикла**  **int sum = 0 ;**  **// Цикл**  **for (int i = 0 ; i < k ; i++ )**  **sum = sum + mas[i]; // тело цикла**  **\*psum = sum ;**  **return \*psum; // возвращаемое значение функции**  **};** |
| **При построении блох-схемы использовать обозначения МУ.** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 2 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **char StrMas[] = "Пример(ААА) строки (А) для подсчета пробелов и Символов А !!! ";**  **int NumA = 0;**  **int NumbSpace = 0;**  **printf ( "Исходная строка: %s\n", StrMas);**  **// Цикл подсчета**  **for (int i = 0 ; (unsigned)i <= strlen(StrMas); i++ )**  **{**  **if ( StrMas[i] =='А' ) NumA++;**  **if ( StrMas[i] ==' ' )**  **{ NumbSpace++; StrMas[i] ='\*' ;};**  **};**  **printf ( "Число символов \"А\" = %2d \nЧисло пробелов = %2d \n" ,NumA , NumbSpace );** |
| **При построении блох-схемы использовать обозначения МУ.** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 8 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **int NumdWords = k+1;**  **char Sentence[180] ="";**  **ErrorText = false; // признак ошибок**  **for (int i = 0 ; i < NumdWords ; i++ )**  **{ for ( k = 0 ; k < (int) strlen((char \*)Words[i]) ; k++ )**  **if (i == 0) // первое слово**  **{ if (iswlower ( Words[i][k] ) && k == 0 ) {ErrorText = true; Words[i][k] = (char ) towupper(Words[i][k]); }**  **else Words[i][k] = (char ) towlower(Words[i][k]); }**  **else // другие слова**  **{ if (iswupper ( Words[i][k] )) ErrorText = true; Words[i][k] = (char ) towlower(Words[i][k]); };**  **if ( Words[i][0] !='\0')**  **{ strcat\_s(Sentence , Words[i]);**  **if ( ( i + 1) < NumdWords) strcat\_s(Sentence , " ");};**  **};**  **// Сообщения об ошибках**  **if ( ErrorText == true ) printf ( "\nБыли ошибки в тексте и они исправлены!");** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - 17 |
| **По фрагменту текста функции или программы построить блок-схему алгоритма и письменно пояснить его работу:**  **struct ListElem{**  **ListElem \* pNext;**  **int ListVal; };**  **// Структура простого списка**  **struct List {**  **ListElem Head; // Голова списка**  **ListElem Tail; // Хвост списка**  **int Count; // Счетчик элементов**  **};**  **void DelLastList( List \* pL ){**  **if (pL->Head.pNext == NULL) return ;**  **ListElem \* pE;**  **ListElem \* pTemp;**  **ListElem \* pELast;**  **if (pL->Head.pNext->pNext == NULL) { pL->Head.pNext = NULL ;**  **pL->Tail.pNext = NULL;**  **pL->Count = NULL;return ; } ;**  **pTemp = &(pL->Head) ;**  **pE = pL->Head.pNext ;**  **while ( pE->pNext != NULL )**  **{ pELast = pE;**  **pTemp = pTemp->pNext;**  **pE = pE->pNext; };**  **pELast->pNext = NULL;**  **pL->Tail.pNext = pELast;**  **pTemp->pNext = NULL;**  **(pL->Count)-- ;};** |
| **При построении блох-схемы использовать обозначения МУ** |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |

|  |
| --- |
| ОП 2020  **Вариант** - |
|  |
|  |
| Примечания: 1. Можно использовать печатную информацию (книги, пособия, справочники) и справочники с компьютера. 2. Нельзя использовать: распечатки с принтера, лекции; мобильные телефоны; планшеты; шпаргалки и т.д. |