Вычеркните пункт, который не является алгоритмической структурой

1. Развилка
2. Следование
3. Присваивание
4. Цикл

Расположите по возрастанию размера стандартных типов данных

1. double
2. int
3. char
4. unsigned int

Вычислите значение выражения, записанного на C: `(5*6/4)%2`

1. 1
2. 3
3. 3.75
4. 0

Что выведет следующий фрагмент программы

```
x=1;
if  (x<0) printf("A"); else
  if (x>1) printf("B"); else
    if(x<-1) printf("C");
```

1. A
2. B
3. C
4. ничего не выведет

Надо написать фрагмент программы, присваивающий переменной *a* единицу, если переменная *x* удовлетворяет следующему неравенству $0 \le x < 1$. Какие из приведенных ниже решений не являются правильными?

1. if (x>=0 & x<1) a=1;
2. if ((x<1) && (x>=0)) a=1;
3. if (x<1) if (x>=0) a=1;
4. if (0<=x<1) a=1;

Следующий фрагмент программы должен вычислять произведение всех четных чисел от 2 до n = 10, но в программе допущена ошибка. Укажите номер строки ошибочного оператора.

```
01    int n, s, i;
02    n = 10;
03    s=0;
04    for (i=2; i<=n; i+=2)
05       s*=i;
06    printf("%d",s);
```

В программе определен массив из N элементов. Что делает следующий фрагмент программы?

```
for (i=0; i<N/2; i++)
  { x=a[i];
    a[i]=a[N-1-i];
    a[N-1-i]=x; }
```

1. Зеркально меняет первую и вторую половину элементов массива
2. Меняет местами четные и нечетные элементы
3. Сортирует массив по убыванию величин
4. Сортирует массив по возрастанию элементов

Следующая функция должна всегда возвращать значение аргумента, который был ее передан в предыдущий вызов (в первый вызов – ноль). Однако в ней содержится ошибка, в результате которой она возвращает всегда 0. Возникло подозрение, что что-то не в порядке в описании переменной save. Как его надо исправить?

```
int prev(int pres)
{  int save = 0;
   int ret;
   ret=save;
   save=pres;
   return ret;    }
```

1. int save;
2. auto save = 0;
3. extern save;
4. static save = 0;
5. register save = 0;

Следующая функция должна подсчитывать разность числа левых и правых скобок в символьной строке. Правильно она ли написана?

```
int parenthesis(char *s)
{ int p=0;
  char *q;
  q=s;
  while(*q)
    switch (*q++)
   {  case '(': p++;
      case ')': p--;
   }
  return p;
}
```

1. Абсолютно правильно
2. Пропущены фигурные скобки, охватывающие операторы p++ и p – –
3. Пропущены операторы break после операторов p++ и p – –
4. Пропущен оператор default в конце оператора switch

При вызове следующей функции возникает серьезная ошибка во время выполнения программы. В DOS компьютер «зависает», а в Windows программа «вылетает». В чем дело?

```
char * scopy(char *s)
{ char *t;
  for (int i = 0; i<strlen(s); i++)
  t[i]=s[i];
  return t;
}
```

1. Не выделана память под строку t
2. Недопустимо присваивать одну строку другой посредством обращения с ними как с массивами
3. Функция не может возвращать указатель на строку
4. Неверно заданы границы цикла for

Какие элементы языка С используются для доступа к отдельным битам?

1. Специальные структуры с доступом к отдельным битам
2. Объединения (union) с наложением данных разного типа
3. Перечисления (enum)
4. Структуры
5. Классы (class)

Что выведет следующая программа?

```
main()
{ typedef struct two { int a;
                       int b; } TWO;
 TWO p;
 TWO *q;
 p.a=1; p.b=-1;
 q=malloc(sizeof(TWO));
 q->a=p.a; q->b=2;
 q=&p;
 printf("%d\n",q->b); }
```

1. Ничего не выведет, в ней есть ошибка
2. 1
3. –1
4. 2

Чему будет равняться значение переменной *x*:

```
int g(int a) { return a*a };

double g(double a)
   { return (a<0) ? -a*a : a*a }

void main() { double x; x=g(2); }
```

1. 4
2. –4
3. 4.0
4. –4.0

Чему будет равняться значение переменных *x*, *y*, *z*:

```
void p(int &a, int &b)
   { int c; c=a; a=b; b=c; };

void q(int *a, int *b)
   { int c; c=*a; *a=*b; *b=c; };

void r(int  a, int  b)
   { int c; c=a; a=b; b=c; };

void main()
   { int x=1; int y=2; int z=3;
     p(x,y); q(&y,&z); r(z,x); }
```

1. 1, 2, 3
2. 2, 3, 1
3. 3, 1, 2
4. 3, 2, 1

Чему будет равняться значение переменной *x*:

```
int x = 2;
void s(int &x) { x=3; }
void q(int x)    { x=4; }
void t()    { int x=5; }
main ( s(x); q(x); t(); }
```

1. 2
2. 3
3. 4
4. 5

Чему будет равняться значение переменной *x*:

```
 main()
   {
float x = 3;
while (x!=0) { x-=1.0/3.0; }
   }
```

1. 0
2. –0.3333333
3. +0.3333333
4. Программа зациклится

Что выведет следующая программа:

```
main()
  { int i;
    int n=0;
    for (i=0; i<10; i++)
    { if (i%2==0) continue;
        n++; }
    printf("%d\n",n);
  }
```

1. 10
2. 0
3. 5
4. 9

Что выведет следующая программа:

```
int g(int a)          { return a*a; };
float g(float a)      { return a*a*a; };
double g(double a)    { return a*a*a*a; }
long double g(long double a) { return a*a*a*a;}

 void main() { printf("%f",g(2.0)); }
```

1. 4
2. 8
3. 16
4. 32

Что выведет следующая программа:

```
class A
 { public: A() { printf("A"); }
 };

 class B : protected A
 { public: B() { printf("B"); }
 };

 class C : public B
 { public: C() { printf("C"); }
 };

main() { C c; }
```

1. ABC
2. CBA
3. C
4. A
5. Ничего

Что выведет следующая программа:

```
class vect
 {
  private:
    double x,y,z;
  public:
    vect(double u=0, double v=0, double w=0)
          { x=u; y=v; z=w; }
    friend double norm(vect a);
 };

double norm (vect a) { return a.x*a.x+a.y*a.y+a.z*a.z;}


main()
{ vect v(1,2,3),u;
  u=v;
  printf("%f",norm(u));
}
```

1. 0.0
2. Недопустимый оператор u=v, операция присваивания не переопределена
3. 14.0
4. Функция norm не будет иметь доступ к секции private

Что выведет следующая программа:

```
 class A {
 public:
    A () { m (); }
    virtual void m() { printf("A"); };

};

 class B: public A
 { public:
    B () { m (); }
    void m() { printf("B"); };
 };


main()
{ B b;
}
```

1. AA 2. AB 3. BA 4. BB