Вариант №03_02

① При определении переменной в языке С нам требуется описание этой переменной, а при объявлении - наоборот, описание не нужно. (Пример: определение: int a = 10; описание: int n;).

② Этикета GNU также автоматически определяет какие части большой программы должны быть перекомпилированы, и выполняет необходимые для этого действия.

③ {}, можно не использовать, когда блок кода состоит из одной строки.

④ Циклы с параметром содержит в себе условие выхода, счетчик и блок кода, а цикл с условием содержит в себе только само условие и блок кода, внутри которого это условие выполняется. Цикл while - цикл с условием.

⑤
```
if (усл.)
{
  блок 1;
}
```

```
if (усл.)
  блок 1;
```

```
if (усл.)
{
  блок 1;
}
else
{
  блок 2;
}
```

```
if (усл)
{
  if (усл 2)
  {
    блок 1;
  }
}
```

```
switch (выраж.)
{
case const1: блок 1;
break
case const2: блок 2;
break;
...
case constn: блок n;
break
default: блок;
break
}
```

усл? выраж.1 : выраж 2

⑥
```
int a = 10;
float b = 8.42;
typedef unsigned long UL;
UL var1 = 3.8;
```

⑦
```
#define
```

⑧
```
long int func(int a, int b, float c);

long int func (int a, int b, float c);
{
  return ((a+b)*(int) c);
}
```

⑨ void* используется как указатель на сырые двоичные данные, не имеющие конкретного типа.

```
void *ptr;
*static_cast<double*>(ptr);
```

⑩
```
#include <stdio.h>
int main (void)
{
  int n;
  scanf("%d", &n);
  int a[n][n];
  for(int i=0; i<n; i++)
    for(int j=0; j<n; j++)
    {
      if((i+j) % 2 == 1) a[i][j]=1;
      else a[i][j]=0;
    }
  return 0;
}
```

(11)
```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
 int a=2, b=3;
 int *ap, *bp;
 ap= &a; bp= &b;
 if(*ap>*bp) printf("%d", *ap);
 else printf("%d", *bp);
 return 0;
}
```

(12) char, int

(13)
```
int fact(int n)
{
  int f=1;
  if(n<3) printf("%d", n);
  else
     for(int i=2; i<=n; i++) f *= i;
  return f;
}
```

Функция с использованием рекурсии каждый раз вызывает сама себя, уменьшая значение i на 1. Когда i становится равным 0, функция завершает выполнение

(14)
```
struct list
{
 int field;
 struct list *next;
 struct list *prev;
};
```

(15) Препроцессор.