

Лекция 4

Подпрограммы
Указатели



Задача. Дана строка длиной до 80 символов.
Подчеркнуть гласные русские буквы.

- Тест.
- Исх. строка: Осень зима весна лето
- Результат: - - - - - - - -

Схема главной функции `main`

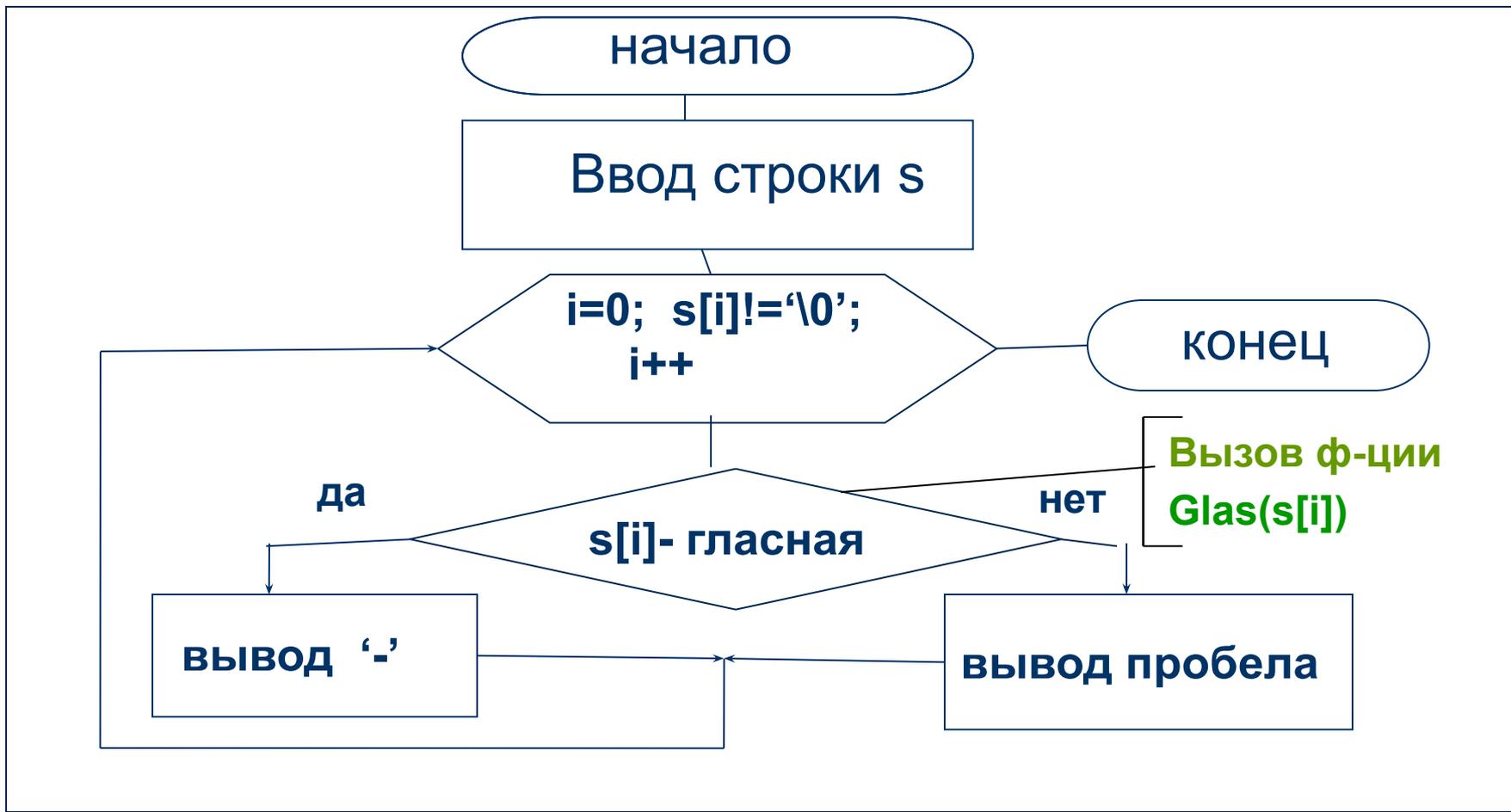
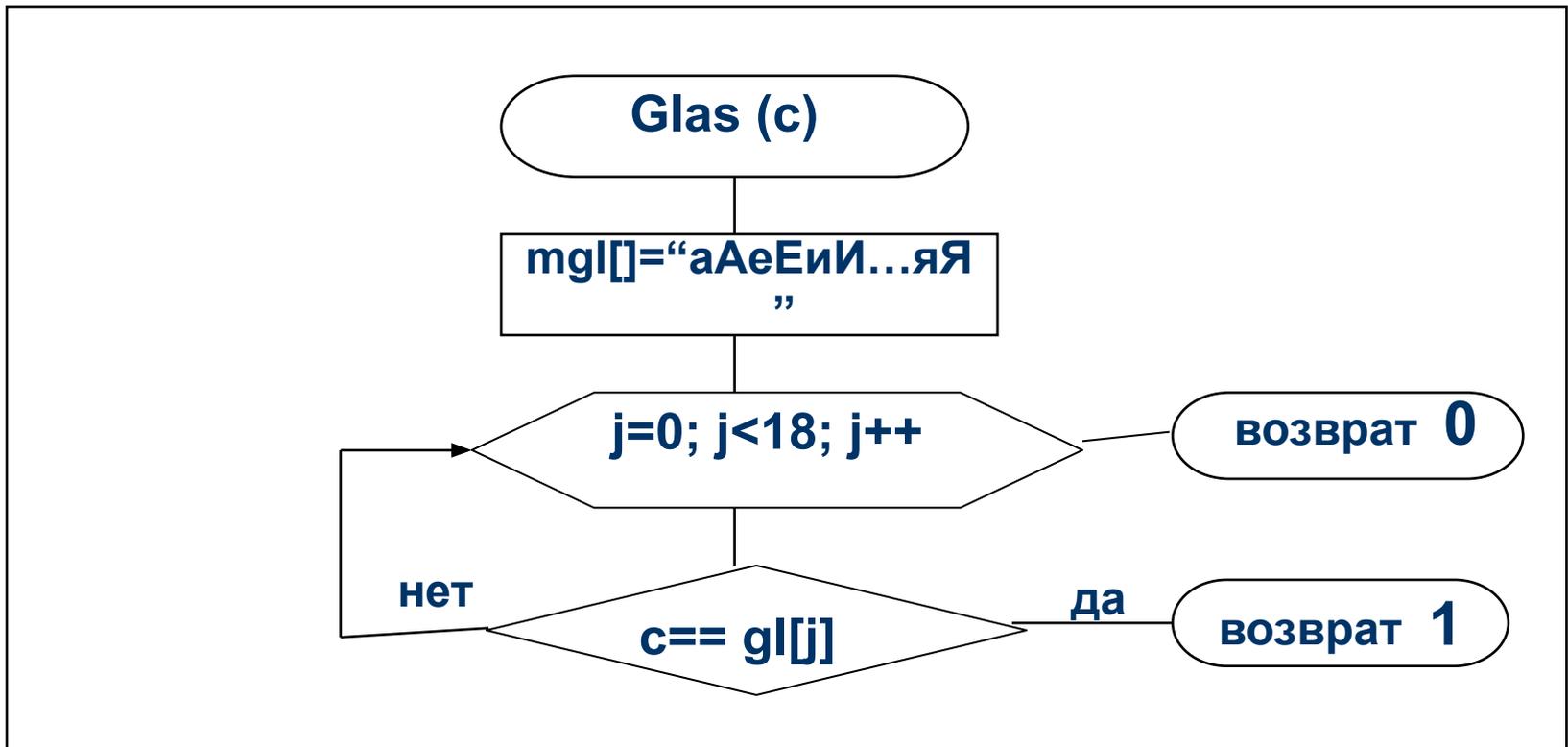


Схема функции ***Glas(c)***, определяющей является ли заданный символ ***c*** гласной буквой.



Программа

```
#include <stdio.h>
    /* Функция проверки, является ли символ гласной
        буквой */
int Glas (char c)
{ char mgl[] = "аАеЕиИ...яЯ"; // массив гласных букв
  int j;
  for (j=0; j<18; j++)
    if (c==mgl[j]) return 1;
  return 0;
}
```

```
void main()
{ char s[80]; // исходная строка
  int i;      // индекс очередного символа строки
  puts ("Введите строку");
  gets (s);
  for (i=0; s[i] != '\0'; i++)
    if (Glas(s[i])) putchar('-');
    else putchar (' ');
}
```

2-й вариант функции *Glas()*

```
int Glas (char c)
{ char mgl[] = "аАеЕиИ...яЯ"; // массив гласных
  // букв

  int j, pr=0;
  for (j=0; j<18 && pr==0; j++)
    if (c==mgl[j]) pr=1;
  return pr;
}
```

Указатели

- **Указатель** – это переменная, которая ссылается на другую переменную. Значением указателя является адрес переменной, на которую он ссылается.
- Описание переменной типа указатель:
ТИП * ИМЯ;

Примеры объявления указателей:

- `int * p1; /* p1 – указатель на цел. переменную */`
- `float * p2; /* указатель на вещ. переменную */`
- `char * p3; /* указатель на симв. переменную */`

- `int n=0, k=1;`
- `int * p1 = &n; /* указателю p1 присваивается
адрес n */`

Пример расположения переменных в памяти:

Адрес	Память	
...	...	
3100	0	n
3102	1	k
3104	3100	p1
	...	

Операция разадресации *

- Чтобы через указатель обратиться к той переменной, на которую он ссылается, используется операция разадресации “*”.

- Пример:

```
int *p1, k=1;
p1=&k;
(*p1)++; /* значение k увеличивается на 1,
          т.к. *p1 - переменная, на которую
          ссылается указатель p1 */
printf (“%d”, *p1); /* вывод значения 2 */
```

Использование указателей в качестве параметров функций

- Пример 1

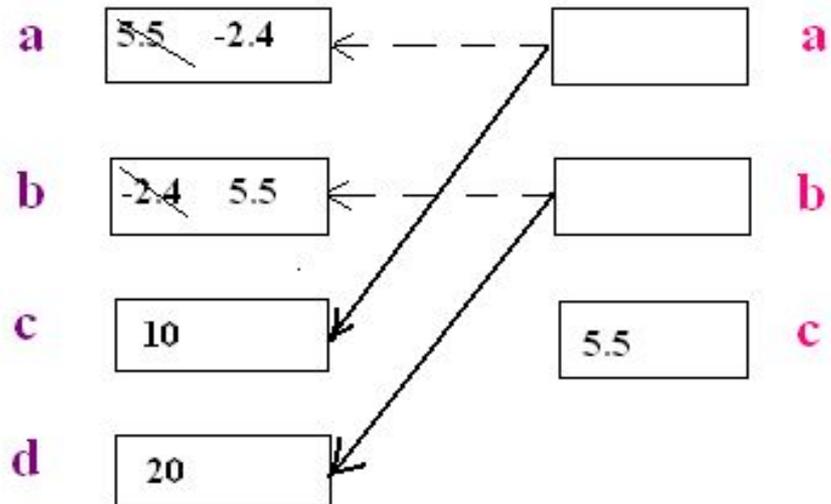
```
/* Функция обмена значений двух переменных */  
void swap (float *a, float *b)  
    /* a,b - указатели на заданные переменные */  
{ float c;  
    c = *a; *a = *b; *b = c;  
}
```

```
/* Гл. функция для тестирования ф-ции swap */
```

```
void main()  
{ float a=5.5, b=-2.4,  
      c=10, d=20;  
  swap (&a, &b);  
  swap (&c, &d);  
  printf ("После обмена a=%.1f, b=%.1f", a, b);  
  printf ("\n c=%.0f, d=%.0f\n",c,d);  
}
```

main:

swap:



Пример 2

```
/* Функция вычисления суммы и количества  
положительных элементов массива */
```

```
void SumPos (float m[ ], int n, float *s, int *k)
```

```
/* Вх. параметры:
```

```
m – указатель на заданный массив,
```

```
n – число элементов массива.
```

```
Вых. параметры:
```

```
*s – сумма положительных элементов массива,
```

```
*k – количество положительных элементов */
```

```
{ int i;  
  for (i=0, *s=0, *k=0; i<n; i++)  
    if (m[i]>0) (*s)+=m[i], (*k)++;  
}
```

/* Пример вызова функции */

float a[20], s;

int k;

...

SumPos (a, 20, &s, &k);