

Ю.Н.Антипов¹, Г.Т.Омаров², Б.К.Шаяхметова³

¹Калининградский технический университет, Россия;

²Карагандинский экономический университет Казпотребсоюза;

³Карагандинский государственный университет им. Е.А.Букетова
(E-mail: kazahzavod@mail.ru)

Средства выполнения домашних заданий по информатике и информационным дисциплинам

Домашние задания имеют большое значение в деле улучшения качества учебно-воспитательной работы по информатике и информационным дисциплинам. Правильно организованные домашние задания обеспечивают прочность знаний студентов, вырабатывают у них навыки самостоятельной работы, развивают у студентов упорство и настойчивость. В настоящей работе рассмотрены эти и другие вопросы, а также конкретные, стандартные примеры.

Ключевые слова: информационные дисциплины, самостоятельная работа, домашние задания.

Проблема выполнения домашних заданий студентами считается вечной в истории высшей школы. Мировой опыт показывает, что формирование личности профессионала любого профиля всецело определяется совершенством организации выполнения им домашней работы. Домашние задания имеют громадное значение в деле улучшения качества учебно-воспитательной работы по информатике и информационным дисциплинам. Правильно организованные домашние задания обеспечивают прочность знаний студентов, вырабатывают у них навыки самостоятельной работы, развивают у студентов упорство и настойчивость.

Программы по информатике и информационным дисциплинам указывают, что «выполнение домашних заданий являются обязательной работой для студентов», и выделяют на каждую тему определенное время для этой работы, в среднем 40–50% того времени, которое предусмотрено для занятий по данной теме. Преподаватель, составляя домашнее задание, должен учитывать время, необходимое для выполнения намечаемой им работы, помня, что чрезмерно большой объем задания заставляет студентов выполнять его наспех, списывать у товарища.

Домашние задания не должны быть чрезмерно однообразными по содержанию, сводиться к выполнению однотипных упражнений по текущей теме. В заданиях рекомендуется включать, кроме текущего материала, решение задач по ранее пройденным разделам программы.

Любое домашнее задание должно быть посильным для большинства студентов, они должны понимать смысл задания и отчетливо представлять, как выполнять его; посильность домашнего задания органически связана с усвоением и пониманием студентами материала, проработанного на занятиях; поэтому преподаватель, намечая домашнее задание, должен ясно представлять, подготовлены ли студенты к его выполнению.

Давая задачи или упражнения, требующие особой сообразительности или применения искусственных приемов для решения, преподаватель должен предупредить об этом студентов, а иногда дать соответствующие указания. Наблюдения показывают, что многие студенты не умеют правильно готовить домашние задания по информатике и информационным дисциплинам; заучивают наизусть определения, выводы и комментарии, стараются в точности запомнить буквенные обозначения (это похвально), но, к сожалению, не умеют связать условия задачи с соответствующим теоретическим материалом. Отсюда заботой преподавателя является учить студентов выполнять домашние задания. Полезно для этого время от времени проводить специальные занятия по обучению работе с учебником, по самостоятельному разбору материала и выводов из него; необходимо к вопросу о порядке приготовления домашнего задания возвращаться часто: и на лекциях, и на практических и на лабораторных занятиях. Домашние задания только тогда достигнут поставленной цели, когда они своевременно проверяются преподавателем.

Приведем некоторые ситуации, возникающие при выполнении домашних заданий.

Итак, 1-й курс, дисциплина «Алгоритмизация и языки программирования», несколько задач, входящих в домашнее задания [1].

Например:

1. Треугольник задан координатами своих вершин. Найти периметр и площадь треугольника.

Для того чтобы решить эту задачу, студенту необходимо напомнить математические формулы нахождения расстояния между двумя точками, заданными своими координатами: если $A(x_1, y_1), B(x_2, y_2)$, то $|AB| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$, и формулу Герона (нахождение площади треугольника по трём его сторонам): если треугольник ABC со сторонами a, b, c , то $S_{\Delta} = \sqrt{p(p-a)(p-b)(p-c)}$, где p — полупериметр треугольника. Эти данные преподаватель под грифом «подсказка» может выписать на доске. Те студенты, которые эти формулы помнят, воспримут их как напоминание, а для непомнящих они будут подсказкой.

Итак, приведенная выше задача может быть решена следующим образом:

Листинг 1

```
Var x1, x2, x3, y1, y2, y3, a, b, c, p, s: real;
Begin
write('Введите коорд. 1-й вершины ->');
readln(x1, y1);
write('Введите коорд. 2-й вершины ->');
readln(x2, y2);
write('Введите коорд. 3-й вершины ->');
readln(x3, y3);
a:=sqrt(sqr(x2-x1)+sqr(y2-y1));
b:=sqrt(sqr(x3-x2)+sqr(y3-y2));
c:=sqrt(sqr(x3-x1)+sqr(y3-y1));
2p:= a+b+c;
s:=sqrt(p*(p-a)*(p-b)*(p-c));
write('Периметр= ', 2p:6:4, 'Площадь= ', s:6:4);
readln;
end.
```

2. Даны вещественные числа x и y . Если x и y отрицательны, то каждое значение можно заменить его модулем; если отрицательно только одно из них, то оба значения увеличить на 0,5; если оба значения неотрицательны и ни одно из них не принадлежит отрезку $[0,5, 2]$, то оба значения уменьшить в 10 раз; в остальных случаях x и y оставить без изменения.

Здесь преподаватель должен использовать подсказку, так как без неё решить дома эту задачу достаточно сложно.

Подсказка:

1. В этой задаче желательно использование сопроцессора для вычислений с плавающей точкой ($\{SN+\}$).

2. Необходимо использование типа `extended`.

3. В задаче необходимо применять булевы отношения.

4. Не один раз использовать условный оператор.

Итак,

Листинг 2

```
{SN+}
Var x, y: extended
Begin
Write('Введите числа x и y (вещественные)');
Readln(x,y);
If (x<0) and (y<0) then
Begin
x:= -x;
y:= -y;
end
```

```

else
if not ((x>=0.5) and (x<=2)) and
not ((y>=0.5) and (x<=2)) then
begin
x:= x/10;
y:= y/10;
end;
write ('После преобразований x= ', x, 'y= ', y);
end.

```

3. Вычислить $S = \sqrt{2 + \sqrt{2 + \sqrt{2 + \dots + \sqrt{2}}}}$.

Для решения этой задачи дома (в виде домашнего задания) преподаватель должен указать подсказку.

Подсказка. Здесь необходимо применить оператор цикла с предусловием, причем его частный случай — цикл со счётчиком, и рассмотреть два варианта: если шаг = 1, то он имеет вид: for K:= K1 to K2 do S, если же шаг=-1, ключевое слово to заменяется словом downto: for K:= K1 downto K2 do S.

Поэтому листинг программы нашей задачи имеет вид

Листинг 3

```

Var n, i: integer;
S: real;
Begin
Write('Введите натуральное число n-> ');
Readln(n);
S:=0
For i:= n downto 1 do
S:= sqrt(2+S);
Writeln('Сумма = ', S);
End.

```

Итак, предлагается следующая задача, которая характеризует некоторые модули и функции.

4. Дано вещественное положительное число $\varepsilon = 10^{-3}$. Вычислить с точностью ε площадь, заключенную между дугами двух кривых: $y = \sin x^2 + 2$, $y = e^{x^2}$.

Подсказка

1. Построить чертеж к задаче на экране монитора.
 2. Для уточнения корней нужно воспользоваться методом приближенного нахождения корней уравнения. Вычислить корни уравнения $y = \sin x^2 + 2 - e^{x^2} = 0$, с точностью $\varepsilon = 10^{-3}$.

3. Использовать подпрограммы.

4. Воспользоваться методом Симпсона для приближенного вычисления определенных интегралов.

5. Применить оператор цикла с постусловием (repeat...until).

Для образца приведём листинг этой программы.

Листинг 4

```

Uses graph;
Var
grdriver, grmode, errcode: i, h: integer;
a, b, e, c, d, x, y, k: real;
i1, i2, h1, m, n, s: real;
function fn(t: real): real;
begin
fn:= sin(sqrt(t)) + 2 - exp(t*t);
end;
function kor(l: real; k: real): real;

```

```

var s: real;
begin
if fn(l)=0 then s:=1
else if fn(k)=0 then s:=k
else
begin
repeat
s:= (l+k)/2;
if fn(s)=0 then s:=l+k*2
else if fn(l)*fn(s)>0
then l:=s
else k:= s1
until abs(k-l)<el
s:= (l+k)/2;
kor:= s;
end;
end;
BEGIN
grdriver:= detect;
initgraph(grdriver,grmode, '');
errcode:= graphresult;
if errcode= grok then
begin
setcolor(5);
line(0, 160, 630, 160);
line(140, 0, 140, 350);
h:= 30;
i:= 20;
repeat
line(i, 158, I, 162);
i:= i+h
until i>630;
h:=40;
i:= 40;
repeat
line(138, I, 142, i);
i:= i+h
until i>=350;
x:=-2;
k:=0.01;
while x<=3 do
begin
putpixel(140+trunc(40*x).
160-trunc(40*(sin(sqrt(x))+2)), 5);
putpixel(140+trunc(40*x).
160-trunc(40*exp(x*x)), 5);
x:= x+k
end;
readlnl
closegraph;
end
else halt(1);
writeln('Укажите точность вычисления пределов интегрирования');

```

```

readln(e);
repeat
writeln('Укажите концы первого отрезка');
read(c, d);
until fn(x)*fn(d)<=0;
a:= kor(c, d)
repeat
writeln('Укажите концы второго отрезка');
read(c, d);
until fn(x)*fn(d)<=0;
b:= kor(c, d);
writeln(a);
a(writeln(b));
writeln('Укажите точность вычисления интеграла');
readln(e);
writeln('Укажите число узлов интерполяции');
readln(n);
i1:= 0;
repeat
n:= n*2;
h1:= (b-a)/n;
s:= fn(a) + fn(b);
x:=a+h1;
repeat
s:=s+4*fn(x)+2*fn(x+h1);
x:=x+2*h1
until x>=b;
i2:= h1/3*s;
m:= abs(i1-i2);
i1:=i2;
until m<=15*e;
writeln('Искомая площадь ', i2);
readln;
END.

```

Теперь приведем задачу домашнего задания в подсказке, в которой нужно подробно описать алгоритм решения.

5. Отсортировать массив, используя метод быстрой сортировки.

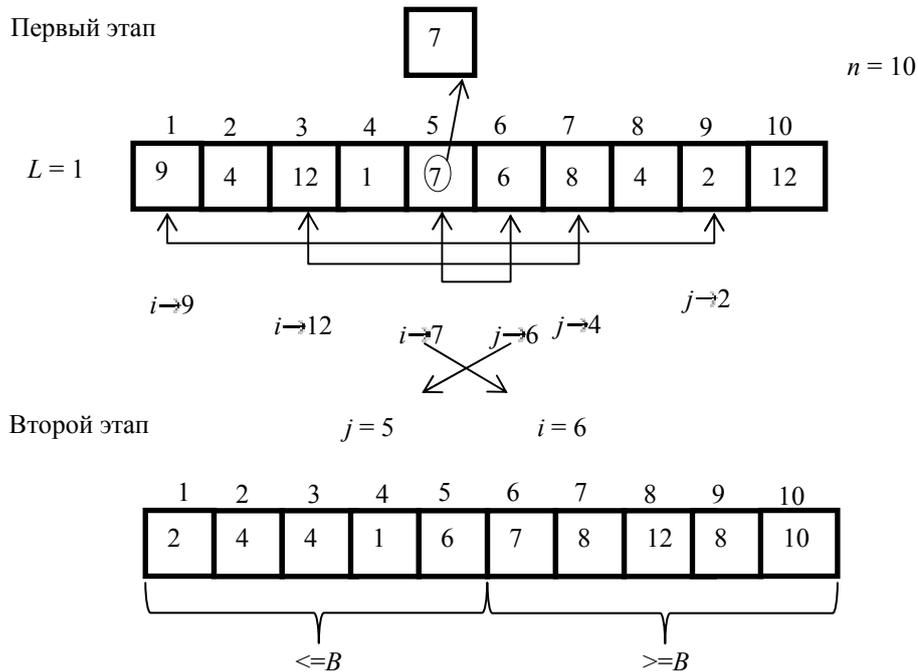
Подсказка.

Метод быстрой сортировки разработан в 1962 г. профессором Оксфордского университета К. Хаором. Принцип метода заключается в следующем.

На первом этапе выбирается центральный элемент массива A и записывается в переменную B . Затем элементы массива просматриваются поочередно слева направо и справа налево. При движении слева направо ищется элемент $A[i]$, который больше или равен B , и его позиция запоминается. При движении справа налево ищется элемент $A[j]$, который меньше или равен B , и его позиция также запоминается. Найденные элементы меняются местами, и встречный поиск по указанным условиям продолжается. И так далее, пока при очередной итерации поиска встречные индексы i и j не пересекутся.

После этого первый этап считается законченным, и элементы исходного массива оказываются разделенными на две части относительно значения B — все элементы, которые меньше или равны B , располагаются слева от пересечения индексов i и j , а все элементы, которые больше или равны B , располагаются справа от границы. Таким образом, относительно значения B массив получается отсортированным, но левая и правая его части не упорядочены.

На втором этапе повторяются действия первого этапа для левой и правой частей массива в отдельности. В результате массив оказывается разбитым уже на четыре отдельно отсортированные части.



На третьем этапе повторяются действия первого этапа в отдельности для каждой из четырех частей и так далее, пока длина сортируемых частей не станет равной одному элементу и, следовательно, все элементы массива будут упорядочены.

Поскольку на каждом этапе повторяются одни и те же действия в разных индексных рамках массива, то их удобно оформить в виде рекурсивной процедуры.

Листинг 5

```

Const n=10;
Var a:array [1:n] of integer;
I:word;
Procedure QSort(L, R:word);
Var B, tmp:integer;
i, j:word;
begin
  B:=A[(L+R)div2];
  i:=L; j:=R;
  while i<=j do
  begin
    while (A[i] < B do
    i:=1+i;
    while A[j]<B do
    j:=1+j;
    if i<=j then
    begin
      tmp:=A[i];
      A[i]:=A[j];
      A[j]:=tmp;
      i:=i+1;
      j:=j-1;
    end
  end

```

```
end
if L<j then QSort(L,j);
if j<R then QSort(i, R);
end;
BEGIN
Writeln ('Введите элементы массива: ');
For i:= 1 to n do
Read (A[i]);
Readln;
QSort(1,n);
Writeln('Отсортированный массив: ');
For i:=1 to n do
Write (A[i]:8)
END.
```

Список литературы

- 1 Гуденко Д.А., Петроченко Д.В. Сборник задач по программированию. — СПб.: Питер, 2003. — 475 с.: ил. (Сер. «Ком Ас»).

Ю.Н.Антипов, Г.Т.Омаров, Б.К.Шаяхметова

Информатика және ақпараттық пәндерде үй жұмыс тапсырмаларының орындаудың әдістері

Мақалада үй жұмысы тапсырмаларына көп мән беріледі. Дұрыс ұйымдастырылған үй тапсырмалары студенттің білімінің жоғарғы деңгейін, олардың өзіндік жұмыстарының нығая түсуін қалыптастырады. Осы және басқа мысалдар аталған жағдайда жан-жақты қарастырылды.

Yu.N.Antipov, G.T.Omarov, B.K.Shayakhmetova

The resources of doing homework on computer science and information disciplines

Homework is important in improving the quality of educational work in computer science and information disciplines. Properly organized homework provides durability of students' knowledge develops their ability to work independently, develops students' persistence. These and other issues, and specific standard examples discussed in present work.

References

- 1 Gudenko D.A., Petrochenko D.V. *Collection of tasks on programming*, Saint Petersburg: Piter, 2003, 475 p.: il. (Ser. of «Kom As»).