

# Проект «Спираль Улама»

Морозов В. В.

В квадратную таблицу чисел размером  $n \times n$  впишем числа от 1 до  $n^2$  по спирали, начиная от центра:

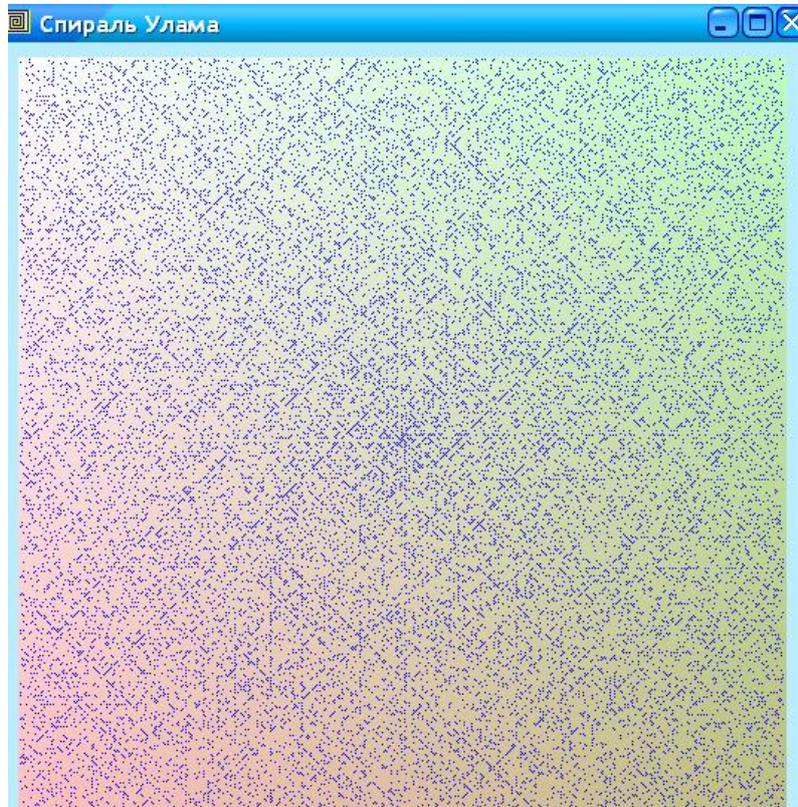
73	74	75	76	77	78	79	80	81	82
72	43	44	45	46	47	48	49	50	83
71	42	21	22	23	24	25	26	51	84
70	41	20	7	8	9	10	27	52	85
69	40	19	6	1	2	11	28	53	86
68	39	18	5	4	3	12	29	54	87
67	38	17	16	15	14	13	30	55	88
66	37	36	35	34	33	32	31	56	89
65	64	63	62	61	60	59	58	57	90
100	99	98	97	96	95	94	93	92	91

Выделим простые числа, то есть числа, которые делятся только на себя и на 1. Число 1 не является простым. Первое простое число – 2.

73	74	75	76	77	78	79	80	81	82
72	43	44	45	46	47	48	49	50	83
71	42	21	22	23	24	25	26	51	84
70	41	20	7	8	9	10	27	52	85
69	40	19	6	1	2	11	28	53	86
68	39	18	5	4	3	12	29	54	87
67	38	17	16	15	14	13	30	55	88
66	37	36	35	34	33	32	31	56	89
65	64	63	62	61	60	59	58	57	90
100	99	98	97	96	95	94	93	92	91

Заметили, что простые числа имеют тенденцию располагаться по диагональным направлениям. Эта конструкция называется Спираль Улама.

Наша задача сегодня – построить спираль Улама как можно большего размера, отмечая простые числа пикселем на форме. Вот что примерно должно получиться.



Ниже приводится технологическая карта проекта «Спираль Улама», с помощью которой учащиеся строят в среде программирования Delphi проект для построения спирали Улама. Предлагаемый проект представляет собой инструмент для исследования асимптотического поведения простых чисел.

1. Запустите Delphi, сохраните новый проект в папке Улам, сохраните файлы проекта как **Ulam.pas** и **SpiralUlam.dpr**.
2. Измените некоторые свойства формы:

name	frmMain
Caption	Спираль Улама
Icon	Поставьте свою иконку
Position	poScreenCenter (это разместит форму по центру автоматически)

На форме не будет никаких других объектов.

3. Введите глобальные переменные

```
var
  frmMain: TfrmMain;
  //Глобальные переменные
  xmin, xmax, ymin, ymax, height: integer;
  xColor, yColor: Tcolor;
  pp: boolean;
```

4. В разделе **implementation** (чуть ниже) введите функцию. Можно потом побаловаться и вводить свою функцию. Эта функция будет определять, является ли ли число простым или нет.

```
implementation
{$R *.DFM}
function proper(c:real):boolean;
  var l:boolean;d,c0:integer;cc:real;
begin
  cc:=c;
  c0:=succ(trunc(sqrt(cc)));
```

```

l:=true;
d:=2;
while l and (d<=c0) do
  begin
    l:=l and (abs(c-(d*int(c/d)))>1e-5);
    d:=d+1;
  end;
proper:=l
end;

```

### 5. Обрабатываем событие FormCreate формы:

```

procedure TfrmMain.FormCreate(Sender: TObject);
begin
  frmMain.height:=570;
  frmMain.Width:=550;
  ymin:=10; ymax:=frmMain.Height-45;
  hight:=ymax-ymin;
  xmin:=10; xmax:=xmin+hight;
  xColor:=clblue;
  yColor:=clAqua;
end;

```

### 6. Обрабатываем событие FormKeyPress формы:

```

procedure TfrmMain.FormKeyPress(Sender: TObject; var Key: Char);
begin
  application.Terminate
end;

```

### 7. Обрабатываем событие FormPaint формы:

```

procedure TfrmMain.FormPaint(Sender: TObject);
var i,j,n,m,c:integer;
begin
  frmMain.Canvas.Brush.Color:=clRed;
  frmMain.Canvas.pen.Color:=clblue;
  for i:=0 to frmMain.Width-35 do
  for j:=0 to frmMain.Height-55 do
    frmMain.Canvas.Pixels[i+xmin,j+ymin]:=rgb(510-(i div 8),510-(j div 8),510-(i+j) div 8);
  n:=hight-2;
  c:=n*n;
  m:=succ(n div 2); //Количество витков
  i:=0;
  repeat
    inc(i); //i-й виток
    for j:=i to n-i+1 do //Верхний ряд i-го витка
      begin
        if proper(c) then
          frmMain.Canvas.Pixels[i+xmin,j+ymin]:=xColor;
          c:=c-1
        end;
    for j:=i+1 to n-i+1 do //Правый ряд i-го витка
      begin
        if proper(c) then
          frmMain.Canvas.Pixels[j+xmin,n-i+1+ymin]:=xcolor;
          c:=c-1

```

```
end;
for j:=n-i downto i do //Нижний ряд i-го витка
begin
  if proper(c) then
    frmMain.Canvas.Pixels[n-i+1+xmin,j+ymin]:=xcolor;
    c:=c-1
  end;
for j:=n-i downto i+1 do //Правый ряд i-го витка
begin
  if proper(c) then
    frmMain.Canvas.Pixels[j+xmin,i+ymin]:=xcolor;
    c:=c-1
  end;
Application.ProcessMessages; //Позволяем системе обрабаовывать другие программы
until i=m;
end;
```

8. **С помощью сочетания клавиш Ctrl+F9 выполните компиляцию программы.** При необходимости исправьте ошибки.
9. **В меню Project выбрали пункт Options.** В отрывшемся диалоге в закладке Application загрузите свою иконку, нажав на кнопку Load Icon. Нажмите ОК. В меню Project выбрали пункт Компилировать. Зайдите в свою папку, найдите откомпилированный exe-файл. Обратите внимание, что файл имеет вашу иконку!! Запустите.
10. **Перепишите (или отксерокопируйте и вклейте) в тетрадь настоящий документ.** Проект завершён.
11. Нарисуйте иконки для формы и приложения, воспользовавшись встроенном в Delphi редактором иконок.