```
const int WIDTH = 120;
const int HEIGHT = 40;
static char screen[HEIGHT][WIDTH];

void initScreen() {
    memset(screen, ' ', sizeof(screen));
}

void putPixel(int x, int y) {
    if (x >= 0 && x < WIDTH && y >= 0 && y < HEIGHT)
        screen[y][x] = '#';
}

void drawLine(int x1, int y1, int x2, int y2) {
    int dx = abs(x2 - x1), sx = (x1 < x2 ? 1 : -1);
    int dy = -abs(y2 - y1), sy = (y1 < y2 ? 1 : -1);
    int err = dx + dy, e2;

    while (true) {
        putPixel(x1, y1);
        if (x1 == x2 && y1 == y2) break;
```

```
void drawRect(int x1, int y1, int x2, int y2) {
    for (int x = x1; x <= x2; x++) putPixel(x, y1);
    for (int x = x1; x <= x2; x++) putPixel(x, y2);
    for (int y = y1; y <= y2; y++) putPixel(x1, y);
    for (int y = y1; y <= y2; y++) putPixel(x2, y);
}

void drawCircle(int cx, int cy, int r) {
    for (int y = -r; y <= r; y++)
        for (int x = -r; x <= r; x++)
            if (x*x + y*y <= r*r)
                putPixel(cx + x, cy + y);
}

void drawEllipse(int cx, int cy, int rx, int ry) {
    for (int y = -ry; y <= ry; y++)
        for (int x = -rx; x <= rx; x++)
            if ((double)x*x/(rx*rx) + (double)y*y/(ry*ry) <= 1.0)
                putPixel(cx + x, cy + y);
}
```

```
class Circle {
public:
    int cx, cy, r;
    Circle(int a, int b, int c) : cx(a), cy(b), r(c) {}
    void draw() { drawCircle(cx, cy, r); }
};

class Rect {
public:
    int x1, y1, x2, y2;
    Rect(int a, int b, int c, int d) : x1(a), y1(b), x2(c), y2(d) {}
    void draw() { drawRect(x1, y1, x2, y2); }
};

class Line {
public:
    int x1, y1, x2, y2;
    Line(int a, int b, int c, int d) : x1(a), y1(b), x2(c), y2(d) {}
    void draw() { drawLine(x1, y1, x2, y2); }
```

```
class Picture {
public:
    vector<Circle*> circles;
    vector<Rect*> rects;
    vector<Line*> lines;
    vector<Triangle*> triangles;
    vector<Ellipse*> ellipses;

    void draw() {
        for (auto c : circles) c->draw();
        for (auto r : rects) r->draw();
        for (auto l : lines) l->draw();
        for (auto t : triangles) t->draw();
        for (auto e : ellipses) e->draw();
    }

    ~Picture() {
        for (auto c : circles) delete c;
        for (auto r : rects) delete r;
```

```cpp
int main() {
    initScreen();
    Picture pic;

    int choice;
    cout << "Enter number of shapes to draw dynamically: ";
    cin >> choice;

    for (int i = 0; i < choice; i++) {
        cout << "\nChoose shape " << i+1 << ":\n";
        cout << "1- Circle\n2- Rectangle\n3- Line\n4- Triangle\n5- Ellipse\n> ";
        int s; cin >> s;

        if (s == 1) {
            int cx, cy, r; cout << "Enter cx cy r: "; cin >> cx >> cy >> r;
            pic.circles.push_back(new Circle(cx, cy, r));
        }
        else if (s == 2) {
            int x1,y1,x2,y2; cout << "Enter x1 y1 x2 y2: "; cin >> x1 >> y1 >> x2 >> y2;
            pic.rects.push_back(new Rect(x1,y1,x2,y2));
        }
        else if (s == 3) {
            int x1,y1,x2,y2; cout << "Enter x1 y1 x2 y2: "; cin >> x1 >> y1 >> x2 >> y2;
            pic.lines.push_back(new Line(x1,y1,x2,y2));
        }
        else if (s == 4) {
            int x1,y1,x2,y2,x3,y3; cout << "Enter x1 y1 x2 y2 x3 y3: "; cin >> x1 >> y1 >> x2 >> y2 >> x3 >> y3;
```

```
Enter number of shapes to draw dynamically: 1

Choose shape 1:
1- Circle
2- Rectangle
3- Line
4- Triangle
5- Ellipse
> 2
Enter x1 y1 x2 y2: 40
5
70
20
```