```c
/* Código final gerado pelo compilador para o programa prime.pas */

#include <stdlib.h>
#include <stdio.h>

typedef struct _f1 {
        struct _f1* parent;
        void* locals[64];
        void* outgoing[32];
        int return_address;
}frame;

typedef enum {
        FALSE, TRUE
} boolean;

int main() {
        frame* fp = NULL;
        frame* sp = NULL;
        int r = 0; /* registo para redirect no final das funções */
        sp = (frame*) malloc(sizeof(frame));

        /* Program name: Primes */

        /* Global Variables */

        int v0; /* i */
        int v1; /* j */
        int v2; /* h */
        int v3; /* k */
        int v4; /* max */
        boolean v5; /* isprime */

        /* Functions */

        /* isqrt() ************************************************** */
        int v7; /* n */
        goto skip_isqrt;
        start_isqrt:
        fp = sp;
        sp = (frame*) malloc(sizeof(frame));
        sp->parent = fp;
        sp->return_address = r;

        int v6; /* isqrt (return value) */
        sp->locals[6] = (int *) malloc(sizeof(int));
        sp->locals[7] = (int *) malloc(sizeof(int)); /* argument: n */
        sp->locals[7] = (int *) v7;

        sp->locals[8] = (int *) malloc(sizeof(int)); /* var: k */
        sp->locals[9] = (int *) malloc(sizeof(int)); /* var: xa */
        sp->locals[10] = (int *) malloc(sizeof(int)); /* var: xo */
        sp->locals[11] = (boolean *) malloc(sizeof(boolean)); /* var: run */

        sp->locals[9] = (int *) 0; /* Assignment */
        sp->locals[10] = (int *) ((int) sp->locals[7]); /* Assignment */
        sp->locals[11] = (boolean *) TRUE; /* Assignment */

        /* WHILE (#7) */
        goto label_while_eval_7;
        label_while_body_7:
        sp->locals[9] = (int *) ((((int) sp->locals[10])+(((int) sp->locals[7])/((int) sp->locals[10])))/2); /* Assignment */

        /* IF (#5) ------------------------------ */
        goto label_if_eval_5;
        label_if_body_5:
        sp->locals[11] = (boolean *) FALSE; /* Assignment */
        goto label_if_end_5;
        label_if_eval_5:
        if ((((((int) sp->locals[9]) >= ((int) sp->locals[10])) && ((((int) sp->locals[9])-((int) sp->locals[10])) <= 1)) || ((((int) sp
        label_if_end_5:
        sp->locals[10] = (int *) ((int) sp->locals[9]); /* Assignment */
        label_while_eval_7:
        if ((((boolean) sp->locals[11]))) goto label_while_body_7;
        label_while_end_7:
        sp->locals[6] = (int *) ((int) sp->locals[9]); /* Assignment */

        v6 = ((int) sp->locals[6]);


        r = sp->return_address;
        sp = sp->parent;
        fp = sp->parent;
        goto redirect;
        skip_isqrt:; /* NOOP is needed because a label can't point to a var dec (eg: int v1;) */

        /* Statements */

        v0 = 1; /* Assignment */
        v4 = 20; /* Assignment */

        /* WHILE (#22) */
        goto label_while_eval_22;
        label_while_body_22:
        v5 = TRUE; /* Assignment */
        v1 = 2; /* Assignment */

        /* calling 'isqrt()' to 'v6' (Call ID: 0) */
        v7 = v0;
        r = 0;
```

```
        goto start_isqrt;
        assignment_0:;
        int c0 = v6;
        v3 = c0; /* Assignment */

        /* WHILE (#17) */
        goto label_while_eval_17;
        label_while_body_17:

        /* IF (#15) ------------------------------ */
        goto label_if_eval_15;
        label_if_body_15:
        v5 = FALSE; /* Assignment */
        goto label_if_end_15;
        label_if_eval_15:
        if ((((v0/v1)*v1) == v0)) goto label_if_body_15;
        label_if_end_15:
        v1 = (v1+1); /* Assignment */
        label_while_eval_17:
        if ((v1 <= v3)) goto label_while_body_17;
        label_while_end_17:

        /* IF (#18) ------------------------------ */
        goto label_if_eval_18;
        label_if_body_18:

        printf("%d", v0);
        printf(" is prime! (1..");
        printf("%d", v3);
        printf(")");
        printf("\n");

        goto label_if_end_18;
        label_if_eval_18:
        if ((v5)) goto label_if_body_18;
        label_if_end_18:
        v0 = (v0+1); /* Assignment */

        /* IF (#21) ------------------------------ */
        goto label_if_eval_21;
        label_if_body_21:
        v0 = (v0+1); /* Assignment */
        goto label_if_end_21;
        label_if_eval_21:
        if ((v0 > 3)) goto label_if_body_21;
        label_if_end_21:
        label_while_eval_22:
        if ((v0 <= v4)) goto label_while_body_22;
        label_while_end_22:

        /* Redirect functions on return */
        goto skip_redirect;
        redirect:
        if (r == 0) goto assignment_0;
        skip_redirect:

        return 0;
}
```